Proc. of the 5th Game-On International Conference, CGAIDE 2004, Reading, UK, pp. 36-40, Nov. 2004.

CASE-BASED PLAN RECOGNITION FOR REAL-TIME STRATEGY GAMES

Danny C. Cheng[1] and Ruck Thawonmas[2]
College of Computer Studies De La Salle University[1]
2401 Taft Ave. Manila Philippines 1004
Email: chengd@dlsu.edu.ph
Intelligent Computer Entertainment Laboratory[2]
Department of Human and Computer Intelligence Ritsumeikan University
Kusatsu, Shiga 525-8577, Japan
Email: ruck@ci.ritsumei.ac.jp

KEYWORDS

Case-based Plan Recognition, User modelling, Real-time strategy game

ABSTRACT

The current game industry around the world is one of the fastest growing industries. One gaming genre that is very popular is the real-time strategy games. However, current implementations of games apply extensive usage of FSM that makes them highly predictable and provides less replayability. Thus, this paper looks at the possibility of employing case-based plan recognition for NPCs so as to minimize their predictability. The paper also looks into possible representation adaptions to minimize the resource requirement to maintain the possibility of deployment in mobile devices.

INTRODUCTION

Applying artificial intelligence in computer games has long been in use starting from the earliest days of such systems (Firclough et. al. 2002) . With the recent rise in popularity of real-time strategy games, it can be noticed that most players of such games prefer to play against human opponents in a multi-player environment as opposed to playing against the computer player. This is due to the fact that minimal effort has been invested into the development and improvement of artificial intelligence in this field due to the enormous amount of overhead both financially and on computing power needed. (Buro and Furtak 2003)

One of the challenges in RTS games is the fact that in RTS games, the worlds normally feature numerous objects, incomplete information, micro-actions, and fast paced actions. Several currently existing works focus mainly on slow-paced, or turn-based games that includes a lot of actions with global effect that would simply overwhelm the human player. (Buro and Furtak 2003)

In this paper, we present some existing works that could be adapted into the area of real-time strategy games. Issues and recommendations are stated at the end of this paper for further development.

REAL-TIME STRATEGY GAMES

Several fields of application and game genres currently exist wherein artificial intelligence research can be applied to. However, this paper focuses on RTS games specifically due to the numerous variety of research problems that exists within the aforementioned game genre. Some research problems would include the following (Buro and Furtak 2003):

- Adversarial real-time planning – planning can take place in several levels namely, strategic, tactical, and operational. Strategic planning would refer to what should be done, tactical refers to how to carry out such plans, while operational would refer to specific actions for each tactical decision. (Kaukoranto et.al. 2003) The problem here is that the environment is dynamic hence pre-defined rules and less than applicable hence alternative approaches have to be investigated.
- Decision making under uncertainty – human players are able to decide on specific plans or strategies even with the lack of information. They are also able to proactively determine the need to look for such information to gain an advantage. Such things might be interesting if they were incorporated to a computer player. (Kaukoranto et.al. 2003)

- Opponent learning and modeling – the ability to determine the players strategies and find ways to react to it in the proper level has been an ideal situation that has been sought after but not yet reached. Most games right now still follow a pre-determined plan of action.
- Spatial and Temporal reasoning – In an ever changing environment, strategies and plans have to be constantly reevaluated for applicability. Understanding of the environment should also be added so as smart decisions can be made.
- Resource management – another task that human players perform is the balancing of allotments for resources. Though recent games are fairly efficient with regards to this, again they are pre-programmed responses to a fixed world environment.
- Collaboration – this is clearly lacking in computer players wherein they never collaborate against a common enemy when attacking it. In contrast, human players usually form teams to fight against a stronger enemy.
- Path Finding – this has always been part of game research since most existing work deal with path finding. The ability to rapidly determine a path in a 2D terrain with moving objects and changing environments has always been a challenge.

In this paper, we look at CBPR (Case-based plan recognition) and some tweaking of the approach as a possible solution to some of the aforementioned research problems focusing more on the user modelling rather than path finding.

CASE-BASED PLAN RECOGNITION

Plan recognition refers to the act of an agent observing the actions of another agent whether it be human or computer-based with the intent of predicting its future actions, intentions, or goals. Several approaches can be used to perform plan recognition namely deductive, abductive, probabilistic, and case-based. It can also be classified as either intended or keyhole. An intended case-based plan recognition system assumes that the agent or the user is actively giving signals or input to the sensing again to denote plans and intensions. In the case of a real-time strategy game, the user or player is focused on playing the game and not focused on trying to convey his or her intention to the sensing agent, hence for this scenario, we would be classified as keyhole plan recognition wherein predictions are based on indirect observations about the users actions in a certain scenario.(Fagan and Cunnigham 2003)

One specific attempt or implementation of case-based plan recognition(CBPR) in games used the game space-invaders as the target platform. (Fagan and Cunnigham 2003) Although this work demonstrated the applicability of CBPR to a certain extent, it also has made several assumptions in its work. First, the set of states are fixed to three, namely Safe(S), Unsafe(U), and Very Unsafe(VU), in a more complex game scenario or genre such as an RTS game, such states may not be finite or defined at the start as they may represent world states at a certain time.

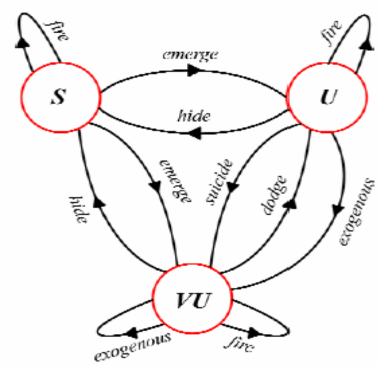

Figure 1. State transition diagram used in the implementation (Fagan and Cunnigham 2003)

EXISTING WORK AND THEIR PROBLEMS

One of the existing work that is applicable to the target domain of real-time strategy games is the work of (Kerkez 2003). The contribution of this work is the presentation of an approach on how to discover and locate plans from incomplete plan libraries.

Most existing work assume that there is a complete plan library to serve as a basis for plan-recognition. However, construction of such a plan-library may not only be not feasible, but the additional or extraneous libraries may affect the performance of the recognizer. (Lesh and Etzioni

1996) Another existing limitation is that "most traditional recognition systems reason in terms of planning actions and do not explicitly keep track of the world states visited during the execution of a plan, except for the initial and the goal states"(Kerkez and Cox 2002).

As illustrated in figure 2, the contribution of (Kerkez 2003)(Kerkez and Cox 2002) is that in the traditional blocksworld problem, the plan representation only incorporated the initial and the end state. The actions that are taken in between and the intermediate states that resulted during the execution was not recorded. This is compared with the representation in figure 2b which is the proposed representation. In contrast, here, the intermediate states are stored for future reference.

Although this approach provides more information and basis for plan-recognition, it also at this point introduced the problem of having too many states to manage and use during recognition. A possible solution as proposed by (Kerkez 2003)(Kerkez and Cox 2002) is illustrated in figure 3.

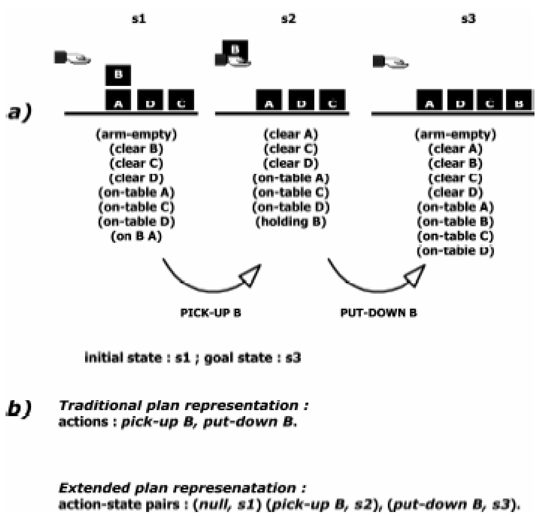

Figure 2. a) An example of a simple planning episode from the blocksworld planning domain. b) Two different views of the observed plan. (Kerkez and Cox 2002)

APPLICABILITY TO RTS GAMES

Based on the aforementioned works from various authors, we believe that there are several considerations needed to be added. The goal is to assist the human player in management tasks in an RTS game such as Warcraft, but not play the game for the user. Although the concepts may be applicable for the opponent NPC, it is not our initial focus.

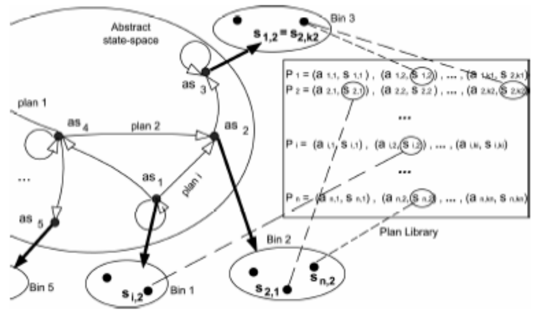

Figure 3. Indexing and storage structures. Abstract states (asi) point to bins (bold lines), containing world states (sj). World states in turn point (dashed lines) to past plans (Pj) in which they are contained. (Kerkez and Cox 2002)

In attempting to apply such methodologies to RTS games, our suggestion is that we limit first the scope of the environment being monitored and controlled by an NPC. This is in order to minimize the possible build up of states that will affect storage and its retrieval specially in limited environments such as a mobile device. An example of a possible limited scope would be in the case of a specific part of the map that contains establishments such as bases rather than the entire map. Hence, strategically, we would be looking at for example defense, or enrichment of resources and fortification rather than plans of attack. Although in (Kerkez 2003), an optimization scheme was suggested based on abstract states indexing and concrete states, it is still not determined if it will be applicable to an RTS game. This is mainly due to the fact that the algorithm may be NP Complete depending on the resulting graph representation of the states. The requirement being that the graph should be either planar or is a circular-arch graph.

Another issue that has to be considered in RTS games is that aside from changing states in the environment, the pieces available or in play can also change depending on the stage of the game being played. As new units are discovered or come into play at higher levels or stages of the game, actions monitored before and subsequent actions taken should be mapped to not only different environment but also different units. A similarity measurement or mapping function may be provided so as to form correlations between what has been monitored before and what to apply now. In this respect we look at hybrid systems like (Schiaffino and Amandi 2000) wherein case-based reasoning which is used to build the cases or plans is combined with bayesian networks to determine the likelihood that a certain action would be peformed. In RTS, this can be viewed as using CBR at strategic and tactical levels to determine similarity features for comparison with other players and bayesian networks can be used to predict the transitions from strategic planning to tactical actions, and then eventually to operational details of the task. Initial set of features for the case-base at each levels is listed in table 1 while table 2 shows a specific example of the case to be stored. The assumption here is that the different lower or more detailed levels are happening based on as a direct consequence of the higher levels decision much like the concept of a chain of command. Temporal information like continous attacks from the opponent that may signify a certain strategy in use though important, is currently not considered in the case-base so as to maintain a level of simplicity at the start. Bayesian networks will be used to determine the subsequent node to be chosen in the next level (figure 4). It will also be used to account for the dynamic world states that could happen in the game such as will the same action be taken for the same scenario and same user profile given the history of the specific user and the variation in the existing types of objects or units present in the current situation.

Plan's determined can be executed at a local or isolated scope or domain so that it would more manageable. The system should also have a means of learning from erroneous predictions. Explicit corrections being made by the player to computer predicted plans and actions taken should be noted so that these can be considered in future attempts at predicting the players possible responses. Much of the considerations here in terms of abstraction and localization is mainly for the purpose of minimizing the resource requirements of the approach. The assumption here is that the client will be able to cache basic abstracted information for initial computation while additional information can be acquired or retrieved as the need arises.

Given all the considerations, the research work aims to perform comparisons based on prototyping and user testing to determine differences or improvements with traditional methods if any. We expect to implement these concepts initially on desktop platforms and then eventually port them to mobile devices such as Palms or PocketPCs. After which, both qualitative and quantitative analysis will be performed to gauge the performance of the system and its scalability and resource requirements. Test deployments on student population would also be included in the testing and evaluation of the results of this work

Table 1. Initial set of sample features at each level of the cases

Levels	***Feature and Description***
Strategic	• Subset of the map for the vicinity in question • Map position • Race and Terrain information • Set of opponents within the vicinity • Set of buildings in the vicinity • Set of friendly units within the vicinity • Gameplay duration
Tactical	• Friendly units available and their capabilities • Resources available • Actions taken
Operational	• Command received • Position of Unit

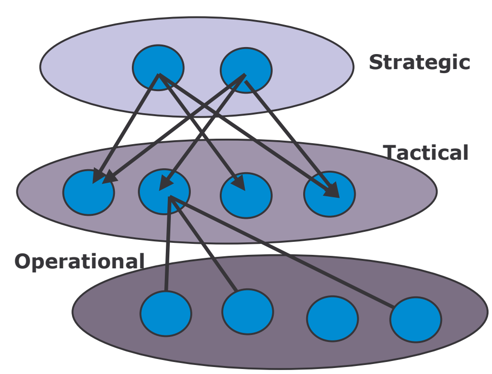

Figure 4. Relationship between the various levels of consideration

Table 2. Sample case entry based on the descriptions using Warcraft as a domain

Levels	*Feature and Description*
Strategic	• 32x32 map unit (size of map) • Lower left corner of map • Human Island Town Terrain • Town hall, Peasants • 0 Enemy Units • 1:30 mins. Into game play
Tactical	• Town Hall Gold: 385 Wood:185 Hit Points: 1500/2400 Armor Type: Fort Armor:5/8 Sight:90/60 Build time:180 • Peasant Gold:75 Wood:0 Food:1 Hit points:220 Armor type: Medium Armor: 0 Sight:80/60 Speed:190 Build time:15 Attack type: Normal Weapon type: Normal Ground attack: 5.5 Air attack:None Cool down:2 Range:Melee • Build Peasant, Militia, Barracks • Set rally point • Scout out enemy position • Send continous attacking militia
Operational	• Peasant: mine gold • Peasant: chop trees • Town hall: build militia • Militia: Wait at rally point • Militia: Move to upper right corner • Militia: Move to upper left corner • Militia: Move to lower right corner • Town hall: build barracks • Barracks: build footman

CONCLUSIONS AND RECOMMENDATION

Currently, the proposed modifications and adaptation have yet to be implemented and tested empirically to determine the appropriateness of the suggestions. However, this research work does present several possibilities that would help improve game play on RTS games on both the desktop and mobile platform. There are additional considerations that are deemed to be ideal inclusions to the research. These would include the detailed study of temporal considerations and the concept of chain of events. In adding this to the research, it would greatly improve the accuracy of the predictions in terms of the plans of a specific user. Also, unlike other games such as an adventure game wherein the goal is either constant or the change is predictable based on the game itself, in a real-time strategy game, the possibility of a change in strategy in the middle of game play is very possible and there is no support structure within the game itself that will aid in the identification of such changes. Hence, issues such as how often should re-evaluation happen comes into view. Also, the current assumption of a strategy is based on a subset map of a certain stage or world in an RTS game. In such events, issues such as complementing or supplementing strategies have yet to be researched on.

REFERENCES

Fagan M., Cunnigham P. 2003, "Case-based plan recognition in computer games". Technical Report TCD-CS-2003-01 Trinity College Dublin Computer Science Department

Fairclough C., Fagan M., Namee B. M., Cunnigham P. 2002. *Research Directions for AI in Computer Games.* Department of Computer Science Trinity College Dublin Ireland.

Kaukoranto T., Smed J., Hakonen H., 2003 "Role of Pattern Recognition in Computer Games". *Proceedings of the 2nd International Conference on Application and Development of Computer Games* p. 189-194

Kerkez, B, & Cox. M. T. 2002. "Case-based plan recognition with incomplete plan libraries". *In B. Bell & E. Santos (Eds.), Proceedings of the AAAI Fall 2002 Symposium on Intent Inference* (pp. 52-54). Menlo Park, CA: AAAI Press / The MIT Press.

Kerkez, B. 2003. "Incremental Case-based Keyhole Plan Recognition." Technical Report #WSU-CS-01-01. Wright State University

Lesh, N., & Etzioni, O. 1996. "Scaling up goal recognition". *In Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning* (pp 178-189).

M. Buro & T. Furtak, 2003 "RTS Games as Test-Bed for Real-Time Research", *Invited Paper at the Workshop on Game AI, JCIS*

Schiaffino S., Amandi A., 2000 "User profiling with case-based reasoning and bayesian networks". *International Joint Conference IBERAMIA-SBIA* pp. 12-21