Search-based planning for virtual agents' behaviour

Miguel Lozano[1], Marc Cavazza[2], Fred Charles[2], Steve J. Mead[2]

[1] University of Valencia
Dr. Moliner 50 (Burjassot) Valencia, Spain.
miguel.lozano@uv.es

[2] School of Computing and Mathematics
University of Teesside, TS1 3BA Middlesbrough, United Kingdom.
{m.cavazza, f.charles, steve.j.mead}@tees.ac.uk

Abstract. This paper presents an attempt to integrate planning facilities as the most generic description of intelligent behaviour for 3D embodied agents.
Applying the state of the art of heuristic search planners (HSP), we present a MinMin approach which is appropriate to control autonomous characters in real time graphics environments.
In this paper we will describe how heuristic search domains can be applied to 3D IVAs as the most generic representation of its intelligent behaviour and also justify the use of MinMin in behavioural animation domains. We will also illustrate the planning functionalities obtained using an extension of the classical dinner-date problem, now carried out in 3D virtual environments.

1 Introduction

The increased graphic realism of virtual agents has generated corresponding expectations on their intelligence. For a virtual agent, intelligent behaviour consist in selecting an appropiate sequence of actions and this can thus be described as a planning problem.

Several approaches have been proposed in this field, probably the most common is using Final State Transition Networks (FSTN) as compiled plans, these however lack flexibility do not make agents goals visible enough. Geib [10] has proposed the use of refinement planning following a detailed study of animation requirements[19][10]. Funge has used situation calculus to generate intelligent behaviours for virtual actors,[9] and Cavazza [6][5] has approach this problem with Hierchical Task Networks (HTNs) for storytelling, considering the *knowledge intesive* nature of this kind of applications.

The planning requirements for virtual actors depend on the specific application, however we can identify these essential requirements:

- The domain representation should be appropiate to embodied agents in virtual environments and identify both goals and physical actions.

– Solution plans should be computed efficiently, considering the time scale of a virtual agent.
– In some cases when the agent evolves in a dynamic environment there is need to interleave planning and execution as well.

There has been recently a renewed interest in search-based planning techniques, as these have demonstrated significant performance on various planning tasks [3][12] [16][18].

This has led us to apply a heuristic search planner (HSP), investigating the integration of fully planning capabilities in virtual actors.

The rest of this paper is organised as follows, firstly we will present the domain representation and the behavioral animation problem proposed to integrate planning within 3D embodied agents. Then we will review the keypoints associated to HSPs and finally, we discuss the results obtained by this integration describing a classical planning problem which is also relevant to a character animation.

2 Planning for 3D embodied agents

As we have previously introduced, several behavioural animation methods have been used traditionally of 3D agents to control the actuation skills and interaction capabilities.

In animation domains, planning capabilities will consist in finding a right sequence of actions that let an agent achieve its goals, with the added value of seeing the solution-plan carried out by a 3D embodied agent. Considering this, planning systems will provide embodied agents with a general method to drive their intelligent behaviours, which will be displayed at the virtual environment in order to see how the agent can solve their virtual planning problems.

In this context, plan optimality will not be an essential requirement, however we will be normally interested in minimum length plans, that is, the minimum sequence of actions that let the 3D embodied agent to achieve its goals. The visualization of these minimum-length plans will display an efficient/intelligent behaviour associated to the 3D agents.

For instance, intelligent agents in simulation systems could compute solution plans in response to user's instructions.

To integrate the mentioned planning capabilites in 3D embodied agents we have considered the example that Figure 1 shows. It is an extension of the classical $dinner - date$ planning problem, where basically an agent must calculate a plan to achieve all the neccesary to prepare a dinner, such as removing the garbage, wrapping a present, etc. We have extended this problem with more operators ($music, computer - work, ...$) but also with new goals and preconditions, such as to finish the work or to require fun for cooking.

Moreover, this scenario has similarities with the storytelling application described in [6] and give us an opportunity to investigate with a (non-decomposable, non-empty delete-lists) planning problem on a similar application.

Problem : Funny -Dinner -Date
Initial_State : garbage, cleanHands, quiet, work
Goal : dinner, present, not garbage, not work

Operator: **Cook** Preconditions: CleanHands, fun, work Effects: dinner	Operator: **Wrap** Preconditions: quiet, cleanHands Effects: pr esent
Operator: **Carry** Preconditions: Effects: garbage, cleanHands	Operator: **Dolly** Preconditions: Effects: not garbage not quiet
Operator: **Music** Preconditions: Effects: fun not quiet	Operator: **Wash** Preconditions: Effects: cleanHands not quiet
Operator: **Phone** Preconditions: cleanHands Effects: fun not quiet	Operator: **Relax** Preconditions: Effects: quiet
Operator: **Paint** Preconditions: quiet Effects: fun not cleanHands	Operator: **Computer -Work** Preconditions: Effects: no t fun not cleanHands garbage not work

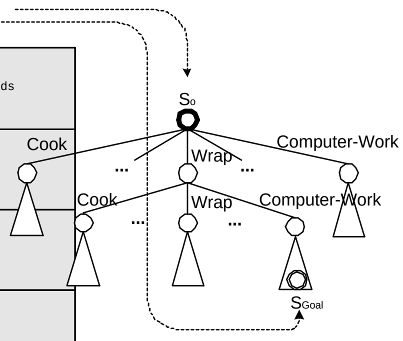

Fig. 1. Search domain representation

As we know HSP domains are represented mainly by three elements: a) the domain representation of the problem, b) the search algorithm and c) the heuristic function. In the next subsections, we will review the integration of these tree key elements in our behavioural animation domains.

2.1 The search domain representation

Although the dinner-date problem was originally formulated in relation with the Graphplan system, it can be easily redescribed in the HSP context using atomic Strips representation (ignoring delete-list for the computation of the heuristic function).

Our agent-centred approach is based on the typical state-model representation for planning domains [3]. Basically, each state will contain a set of atoms representing the agent state (see Figure 1, e.g. *(cleanHands, not garbage, not work, ...)*). To complete the problem formulation the agent will require a set of operators which will represent its effectory capacity, mapping states to successor states according to its preconditions.

In the context of planning for behavioural animation, we must notice that the quality of the plans will be directly related to their lengths, so that, the agents must avoid long-length plans which will display an undesirable behaviour and hence, they should search for plans with the minimum-length associated. Longer plans are often non optimal in their action sequence. For instance, an agent who wash his hands before carrying out the garbage, will have to wash his hands again.

To achieve this we are using the classical depth bounding, d criteria, which will stop all the plans calculations beyond the maximum length plan allowed d. In this way, the depth level reached by a goal state of any plan-solution, will represent the length of this plan and it will be the quality factor considered on the final agent decision making at the root of the tree. Considering that the embodied agents should achieve their goals through plans with no actions repeated, we have initialized d as the total number of operators the agent can apply.

Figure 2 shows an scheme of the system architecture where basically the 3D emodied agents can apply their plan operators as a atomic actions carried out in the virtual environment.

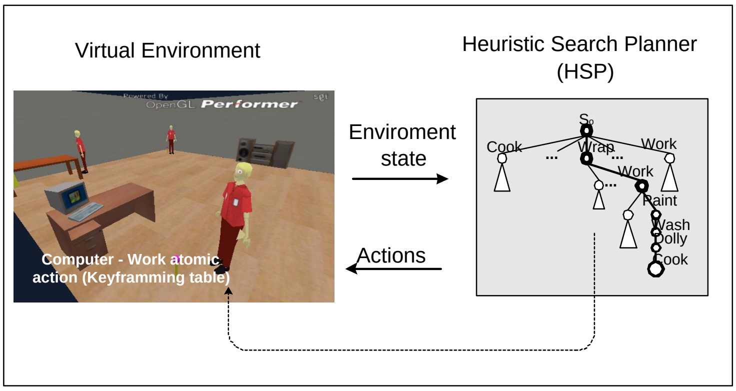

Fig. 2. System Architecture

Taking into account the domain representation introduced, the next subsection will present MinMin as an adequate search algorithm to supply the planning requirements for our 3D embodied agents.

2.2 Planning with MinMin

MinMin [12] has been proposed as a search algorithm for real time decision taking. It has the advantage of searching forward from the current state to a fixed depth horizon and then computes the heuristics values for the frontier nodes. Furthermore MinMin provides a forward search method able to interleave planning and action execution, and to extract the minimum-length plans required.

As Geffner pointed out, [3] the heuristics calculation associated to every node in classical HSPs, is the most expensive computational step associated to HSPs, and MinMin reduces this calculation to the search horizon nodes.

Figure 3 shows how MinMin is capable of refining its solutions during the search using a dynamic depth-bounding criteria. According to this, during the

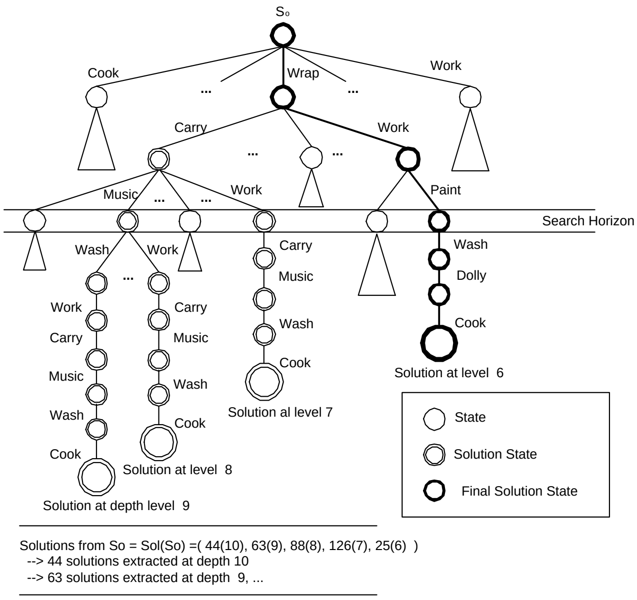

Fig. 3. MinMin search

plan-search this bounding factor d should be decreased to the length of the last best plan extracted (the minimum one). This bounding is also useful to overcome the MinMin main problem, that is cycling. A second bounding criteria has been introduced to MinMin in order to improve its efficiency. This second bounding (2-B) simply detects when a new state with no new effects is going to be created and thus prunes it. (e.g. $S_o - Carry - S_{useless}$, $S_o - Relax - S_{useless}$). The performance of the whole planning system at the $funny - dinner - date$ problem introduced will be shown, as the rest of tests, in the results section.

MinMin control is also adequate to extract the shortest-length plans, though not always the optimal one, as each node will select the child with the minimum cost (i.e. the node which could be part of a minimum length-plan solution). In this way, at the root node tree the agent can perform an informed action selection mechanism, deciding at each plan step the shortest strategy or sub-plan which let him to achieve his goals.

2.3 Heuristics

We are using the independent domain heuristics presented by Bonet&Geffner in [3] which can be easily adequate to MinMin search domains. Heuristics are computed from the horizon nodes by ignoring *delete-list* and expanding the atomic facts that belong to post-conditions until all the atomic facts corresponding to the goal are met.

As mentioned before, the planning system which will control the 3D embodied agent will be interested in plans with a minimum length, so, once a goal state is achieved the heuristic function will simply return the depth of this goal state.

It should be noted that we are not considering any kind of spatial knowledege information at this point, such as, the physical distance to apply the operators, and so, any minimum-length plan will be considered as an acceptable plan-solution for our 3D embodied agents.

Then MinMin search control will select recursively the best child up to the root of the tree, so that, the agent will be informed about the minimum-length plan to carry out from its state and according to it, the next action to perform.

3 Results

The system has been fully implemented and tested over a number of initial configurations in a graphic environment corresponding to the *funny* dinner-date problem.

The table below shows a comparation between MinMin with a simple depth bounding (d=10) and adding the second bounding introduced (2-B).

S. Horizon	Nodes (2-B)	Time (2-B)	Nodes	Time
3	88261	1.39 sec.	215540	3.47 sec.
4	105208	1.65 sec.	383532	6.16 sec.

Table 1.- MinMin results and optimised implementation.

As table 1 shows, the overall performance obtained by MinMin is adequate to 3D real time graphics environments. Furthermore, restricting at S_o the depth-criteria ($d = 7$), MinMin is able to obtain similar 6 plan length solutions in 0.082 secs.

Figures 3,4 illustrate the search-plan carried out by MinMin to solve the *funny dinner-date* problem presented previously. To show the behaviour of MinMin algorithm, we have associated a solution vector to each search state, which indicates the number of solutions extracted by MinMin in several depth levels.

In this way, the agent will start searching from its initial state S_o using MinMin, and will obtain solution plans of length 10. As mentioned earlier, MinMin will refine its solutions during the searching process so when MinMin find a solution shortest than than the current best one it will decrease its depth bounding d. Searching in this way, MinMin finally obtains plans of length 6 (see Figure 3), as the minimum ones. Then the agent at the top of the tree will try to apply the first operator associated to the last minimum plan calculated (e.g., $S_o - wrap - S_1$).

To achieve this in virtual environments, we have associated ($action, position, orientation$) tuples to each operator the actor can apply (Figure 1). The reason

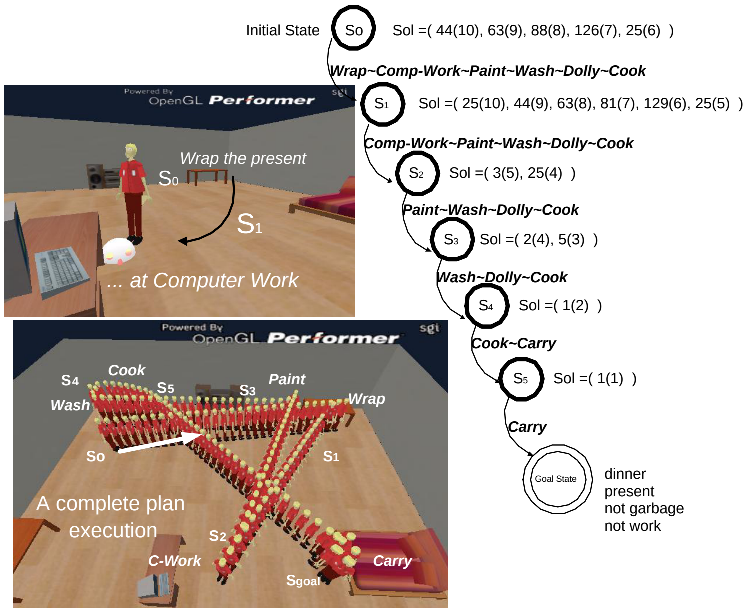

Fig. 4. Integrating search-based planning in 3D virtual environments

is to apply an operator at the virtual environment consisting on reaching the right (*position*, *orientation*) and then executing the atomic-animated *action* associated, typically a simple keyframing table. This couple action-selection in planning, to low-level motion simulation for executing the action.

As figure 4 shows, once the 3D agent has reached the right *position* and *orientation* for wrapping the present, it will execute the associated action (*wrap*). As we are considering only valid operators (those who will add new effects to further states), once the action associated to the operator has been applied, (i.e. the 3D agent has finished its keyframing table) its internal search state would also be modified ($S_1 = (S_o) + present$). Then, the agent must perform a new search from its new state (S_1), interleaving in this way planning and action execution, and achieving finally an intelligent autonomous behaviour able to reduce the distance to its goals.

Finally, we show a complete agent plan execution performed in the 3D virtual environment. To illustrate this, we have overdrawn the agent states and also the operators undertaken at simulation time, indicating the associated actions-positions in the virtual environment. In this way, we can visualise a complete plan solution that will finally represent the intelligent agent behaviour in its 3D virtual environment.

All the tests and results showed in this paper have been performed on a Pentium-IV(1.6Ghz) PC, using Performer graphics API under Linux O.S.

4 Conclusions

We have described a specific approach to integrate fully search based planning capabilities in 3D embodied agents. Performance of the planning system has shown good potential for scaling-up on simulation tests. Our future work will be oriented to include sensing and uncertain information in the planning system, so that, a complete intelligent virtual agent architecture could be tested in 3D virtual enviromental simulations.

References

1. N. I. Badler, C. B. Phillips, and B. L. Webber. *Simulating Humans : Computer Graphics Animation and Control.* Oxford University Press,1993.
2. Norman I. Badler, Bonnie L. Webber, Jugal Kalita, and Jeffrey Esakov, editors. *Making them move: mechanics, control, and animation of articulated figures.* Morgan Kaufmann, 1991.
3. Bonet B, Geffner H. *Planning as Heuristic Search: New results* Proceedings of ECP'99, pp.360-372.
4. Bonet B, Geffner H, Loerincs G. *Planning with incomplete information as heuristic search in belief space.* Proceedings of the 5th. Int. Conf. on AI Planning and Scheduling, Colorado 4/2000.
5. Cavazza M, Charles F. Mead, S. *Agents's interaction in virtual storytelling.* Proceedings of Third International Workshop on Intelligent Virtual Agents 2001. Madrid Spain
6. Cavazza M, Charles F. Mead, S. *Interacting with virtual characters in Interactive Storytelling.* Proceeedings of the Autonomous Agents Conf., AAMAS'02. Bologna, Italy
7. S. Donikian. *Multilevel Modeling of Virtual Urban Environments for Behavioural Animation.* Proc. Computer Animation 97, Geneva, Switzerland, IEEE Press 1997.
8. Elliott C., Rickel J, Lester J. Lifelike pedagogical agents and affective computing: An exploratory synthesis Artificial Intelligence Today. Springer 1999.
9. Funge, John. Cognitive Modelling for games and Animation Communications of the ACM, Julio 2000. Vol 43. no.7.
10. Geib, C., 1994. *The intentional planning system: Itplans.* Proceedings of the 2nd Artificial Intelligence Planning Systems Conference, pp. 55-64.
11. Grand S Cliff D. *Creatures: Entertainment software agents with ai.* Autonomous Agents and Multiagents Systems,1998.
12. Korf, R.E., 1990. *Real-time heuristic search.* Artificial Intelligence, 42:2-3, pp. 189-211.
13. Laird J.E., Nowell A., Rosenbloom P.S. Soar: an architecture for general intelligence Proceedings of the First Intl. Conference on Multiagent Systems. April 1995
14. McDermott, D. *A heuristic estimator for means-ends analysis in planning.* Proc. of the Third Int. Conf.on AI Planning Systems, 1996.
15. Pearl, J., *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Reading (Massachusetts), Addison-Wesley, 1984.
16. Pemberton, J.C. and Korf, R.E., *Incremental Search Algorithms for Real-Time Decision Making.* Proceedings of the 2nd Artificial Intelligence Planning Systems Conference (AIPS-94).

17. Tu,X.Terzopoulos, D. Artificial fishes: physics, locomotios, perception, behaviour. In Computer Graphics (SIGGRAPH94 Proceedings), pages 4350, Orlando, Florida, July 1994.
18. Tsuneto, R., Nau, D. and Hendler, J., *Plan-Refinement Strategies and Search-Space Size.* Proceedings of the European Conference on Planning, pp. 414-426.
19. Webber, B., Badler, N., Di Eugenio, B., Geib, C., Levison, L., and Moore, M., *Instructions, intentions and expectations.* Artificial Intelligence J. 73, pp. 253-269.
20. Weld, Daniel S. *Recent advances in AI planning.* AI magazine, October 1999.