Firefighter Training Based on HTN Planner: from Domain via Hierarchy to Task Fulfillment

Lianchao Li[1] and **Dieter Wloka**[2]

Abstract. Serious games play more and more important roles in current training domains, such as firefighter training in Road Traffic Collision (RTC) scenario, which reduce the cost and time comparing with firefighter drill with real models. On the other hand, a well designed autonomous Non-Playable Characters (NPCs) system controlled by Artificial Intelligence (AI) provides players a better game experience in video games and intelligent assistant in serious games. Hierarchical Task Network (HTN), a well-known automated planning approach in AI, is wildly used in not only video games, but also serious games, and is designed to capture the real life hierarchical structure of the planning domain. However, there are rare research which take this advantage of the game AI to improve the training pattern, especially in firefighter training domain which contains a rigorous hierarchical structure. This paper studies the firefighter domain with a specific task: vehicle stabilization, and introduces an HTN approach to create autonomous NPCs that uses the build-in hierarchy of the firefighter system.

1 INTRODUCTION

Serious games—digital games provide a special game experience for purposes other than mere entertainment—is not a new concept recently in the area of professional education, as training processes are performed in serious games with realistic virtual environments which contain no danger and reduce costs and time comparing with the real-life training method. Emergency management and firefighter training in predefined incident scenarios using serious games technology become more popular and have achieved a great success [11] [13]. Although the continues thrive of the development of those serious games in firefighting domain improves the training efficiency, there are rare experiments on integrating the idea of using AI in serious games to improve the training pattern. Well-designed autonomous NPCs can not only replace the missing team members in not full attendance situation, but also provide a customized role-based training pattern with intelligent assistant.

Firefighter training system in incident scenario for RTC is a category of a special professional education which consists of several sub-roles training in a firefighter group: firefighter leader (or group leader), tool operator, casualty carer and back-up staff (or logistician) [1]. Different roles in incident scenes follow different Standard Operating Procedure (SOP) [1][2][3] and tasks and plans are integrated into the build-in hierarchical structure in firefighter domain [1] as presented in figure 1. In a firefighter group, the firefighter leader is responsible to the overall plan and issues orders to different firefighter squads. Each squad should perform one or many compound tasks and decompose tasks to several sub-tasks, which will be assigned to squad members. Those members will perform a sequence of actions to fulfill the task and report the status and result up until the firefighter leader, who will accordingly make further decisions or plans to optimize the overall plan for rescuing the casualties and protecting the property.

In AI, the HTN is a planning technique that generates plans to solve the specific goal tasks in a hierarchical structured task network, which suits for this firefighter training domain where tasks are naturally organized in a hierarchy [10]. [12] [6] [9] are several examples of using HTN in the area of rescue and firefighter training, but none of them have a deep study of using the build-in hierarchical structure of firefighter domain to create a HTN-based autonomous NPCs system.

We firstly study the firefighter domain and specify the requirement from the firefighter in the RTC problem. Secondly, we introduce an approach to create the HTN-based autonomous NPCs and represent the knowledge according to the domain study. Then, we discuss the result in two aspects: effectivity and ease of knowledge representation. In the end, we summarize our work and discuss the future work.

2 DOMAIN STUDY

Firefighters rescuing trapped persons in an RTC scene is not a general domain used in video games, namely the tasks are not simply attack, defense, cover etc. On the contrary, firefighters perform more complicated tasks and follow more dynamic plans. Before illustrating our approach, this chapter presents our study about firefighter domain in RTC problem and the requirement to solve the problem. Moreover, since it is difficult to present all the tasks performed by firefighter groups, we will take vehicle stabilization as an example to elaborate our HTN-based AI system.

2.1 Scenario simulation

This paper aims to build a HTN-based AI system in an educational serious game which the firefighters rescuing a trapped casualty in a RTC scene so that the study case described in [4] is appropriate as a test scenario. One Mercedes and one VW Golf collided at a high speed head-on during overtake in the forest area, which heavily deformed both cars. One casualty, trapped and heavily injured in the VW Golf, was diagnosed as a polytrauma which was a mortal danger if he was not released and rescued immediately. Therefore, the main task for the firefighter group was to release and rescue the trapped casualty.

[1] University of Kassel, Germany, email: lianchao.li@uni-kassel.de
[2] University of Kassel, Germany, email: dwloka@uni-kassel.de

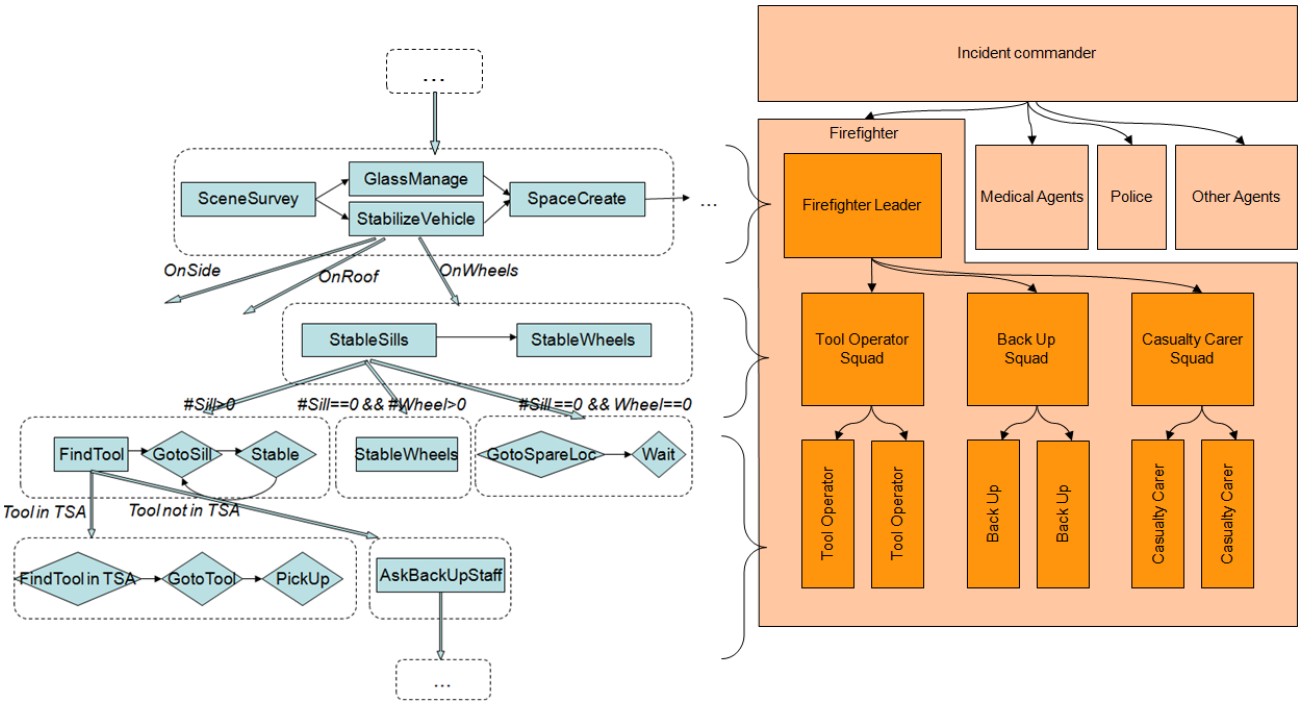

Figure 1. The main hierarchical structure: the right part presents the build-in hierarchical structure in firefighter domain in RTC scene; the left part illustrates the hierarchical task network to solve the StabilizeVehicle problem. The decomposed tasks in HTN matches the build-in hierarchy in firefighter domain. In the HTN part, rectangles present the compound tasks and the diamonds present the primitive tasks; the curved rectangles with dashed line present a method that decompose the compound task where the blue arrow point. Tools are normally exhibit in the Tool Staging Area (TSA), otherwise, backup staff is responsible to get the tools.

2.2 Domain knowledge

As mentioned before, we present a representative example—vehicle stabilization—to elaborate how HTN-based NPCs works in firefighter domain. The vehicle must be completely stabilized in order to prevent further injuries from the casualty and to protect the rescuers before beginning their work. Although a damaged vehicle may be found in many poses after an RTC incident, the features most resulting in the need for stabilization are vehicle resting on wheels, side and roof. Different mechanisms will be used to accomplish vehicle stabilization task according to where the vehicle resting, and we focus on the first condition—stabilize the vehicle while it is resting on wheels. As figure 2 presented, there are two elements need to be stabilized, stable wheels as the lower-left image presented, and stable sills as the lower-right image presented. Stable wheels is to chock the wheels with appropriate wedges and blocks to eliminate any lateral movement, and stable sills is to block the rear of each sill around the jacking points. To stable wheels, a minimum requirement is to stable one wheel, but preferably two - diagonally, which is considered as the best practice. On the other hand, the best practice is to stable 4 points of the sills around the jacking points, which is presented in the upper image of figure 2.

2.3 Command Hierarchy

The structure of organizing the car incident is presented in the right part of the figure 1. Incident commander plays the major role of commanding and planning all agents in an incident scene. Generally, after assessing the extent of the incident, the number of the resources, hazards and risks, the incident commander develops and implements an appropriate strategic plan taking into account the SOP and Dynamic Risk Assessment (DRA) [2]. All agents follow the plan and operating their tasks.

Firefighter leader, as the commander of the firefighter group, assigns tasks to each squads [8]: tool operator squad, back-up squad and casualty carer squad, and members in these squads are performing subtasks individually or cooperatively. For instance, the firefighter leader commands the tool operator squad to stabilize the vehicle and the two squad members will stabilize the vehicle based on the "Crew Approach", which increases the efficiency of the rescue process [2][1]. The "Crew Approach" allows the tool operators simultaneously performing different sub-tasks: stable wheels and stable sills, to save the time. Subsequently, individuals will perform the action sequence to accomplish the according sub-tasks, e.g. find the appropriate tool, go to pick the tool and use the tool correctly to stable the vehicle. On the other hand, individuals provide the result and feedback of the tasks which either successfully accomplished or failed to the squad and then the squad reports to the firefighter leader. In the end, the firefighter leader can generate further tasks or re-plan according to the current result. This naturally hierarchical structure is brief and clear to be implemented in HTN planner.

3 HTN-BASED AUTONOMOUS NPCS

HTN is an automated planning method in AI to search a plan (an action sequence) in the form of a task network to accomplish the goal task via recursively decomposing compound tasks using predefined methods and executing primitive tasks via predefined operators. Tasks in HTN are performed according to the world states that are initialized: preconditions must be satisfied before executing the tasks

Figure 2. Stabilization of a vehicle resting on its wheels [1].

and the corresponding world states should be updated after the tasks accomplished. HTN planning has two types of tasks, primitive tasks and non-primitive tasks. Primitive tasks present in the leaves of the task network, and can be directly accomplished when their preconditions are satisfied. Non-primitive tasks, also named as compound tasks, need to be decomposed into further tasks until reach to the leaves of the task network in order to accomplish them.

3.1 Firefighter domain knowledge representation

HTN, as a domain-independent planner, applies to various real world domains which contain natural hierarchical structures, such as firefighter rescuing domain in RTC scenarios. A planning domain defines methods and operators that are used to perform primitive and non-primitive tasks. Operators are actions or action sequences that directly accomplish primitive tasks. For example, "navigate to a position" is an operator which allows the agent to navigate to the target position in the virtual world. Methods are used for decomposing non-primitive tasks into primitive ones. To decompose a non-primitive task, methods provide several branches. Each branch implements a way of decomposing the non-primitive task into a sequence of tasks or actions. The planner checks whether the preconditions are satisfied according to the current state of the game world before executing a method branch or an operator. For instance, firefighters should stabilize the crashed vehicle before performing the rescuing process. However, according to how the vehicle is positioned after the collision, different method branches should be chosen to stabilize the vehicle, such as stabilization of a vehicle on its tires, side or roof, and each method contains several working steps to fulfill the task [1]. Subsequently, executing tasks and operators can change the state of the world by adding and deleting a list of states. Each method branch leads to a task list, which also contains primitive tasks and non-primitive tasks. Recursively performing the methods until reach to the leaves of the task network, namely no compound tasks are in the task sequence. Therefore, the existence of a linear sequence of operators which could accomplish the goal task indicates that a final plan is found by the HTN planner.

3.2 Knowledge representation

Vehicle stabilization is the example we have studied in chapter 2, and here we will give a detailed explanation of how does our AI system perform this task. The way of representing tasks follows the style of Troy's work presented in [7].

As the left part of figure 1 presented, the firefighter leader performs the overall plan for the whole group. `StabilizeVehicle`, one of the task in the task network, is assigned to the tool operator squad. There are several methods to decompose this compound task according to where the crashed vehicle stands, for instance, `OnWheels` is one of the methods that decompose the non-primitive task `StabilizeVehicle` into two sub-tasks `StableSills` and `StableWheels` by the tool operator squad. The tool operator squad will assign these sub-tasks to each tool operator. The Crew Approach allows the firefighters to perform their tasks simultaneously, two tool operators start to perform each task respectively.

```
Compound task[StabilizeVehicle]
  Method[VehicleResting
           == VehicleOn.Wheels]
    SubTasks[StableSills(),StableWheels()]
  Method[VehicleResting == VehicleOn.Side]
    SubTasks[...]
  Method[VehicleResting == VehicleOn.Roof]
    SubTasks[...]

Compound task[StableSills]
  Method[StablePointSills.size> 0]
    SubTasks[FindToolForSillStable(),
             NavigateToTool(),
             PickUpTool(),
             NavigateToSill(),
             UseToolToStableSill()
             StableSills()]
  Method[StablePointSills.size == 0,
         StablePointWheels.size > 0]
    SubTasks[StableWheels()]
  Method[StablePointSills.size == 0,
         StablePointWheels.size == 0]
    SubTasks[NavigateToSpareLoc(),
           WaitingForOrder()]
```

There are three methods to decompose the `StableSills` task according to how many stable points of wheels and sills remain. When the `StablePointSill` array is not empty, tool operator will stable one of the sill (first element from the array). If sills are all stabled, namely the array of `StablePointSill` is empty, but not the `StablePointWheels` array, the tool operator will help the other tool operator to stable the wheels by performing task `StableWheels`. Otherwise, the tool operator will go to the spare location and waiting for further command from the firefighter leader after all the stable work accomplished.

To accomplish the task by the first method, the tool operator will find the tool to stabilize the sill whose location is stored in the array `StablePointSill` by performing the sub-tasks: `FindToolForSillStable`, `NavigateToTool`,

`PickUpTool`, `NavigateToSill`, `UseToolToStableSill`. `StableSills` task in the end ensures all sills should be stabilized. Recursively performing the tasks `StableSills` and `StableWheels` by two tool operator members respectively until all four sills in the `StablePointSill` array and two wheels in the`StablePointWheels` array are stabled, a succeed message will be sent up until the firefighter leader. And the leader can assign them further tasks according to the state of the current scene, such as SpaceCreate etc. to reach the final rescue goal.

4 RESULT

There are two aspects of result of our HTN-based autonomous NPCs are considered: the effectivity that whether the autonomous NPCs fulfill the task, and the ease of knowledge representation that how intuitively converting the natural hierarchy from the firefighter domain to the hierarchical task network.

4.1 Effectivity

Figure 3 presents the result of successful vehicle stabilization, that all the wheels and sills are stabled with the according tools. The fire-

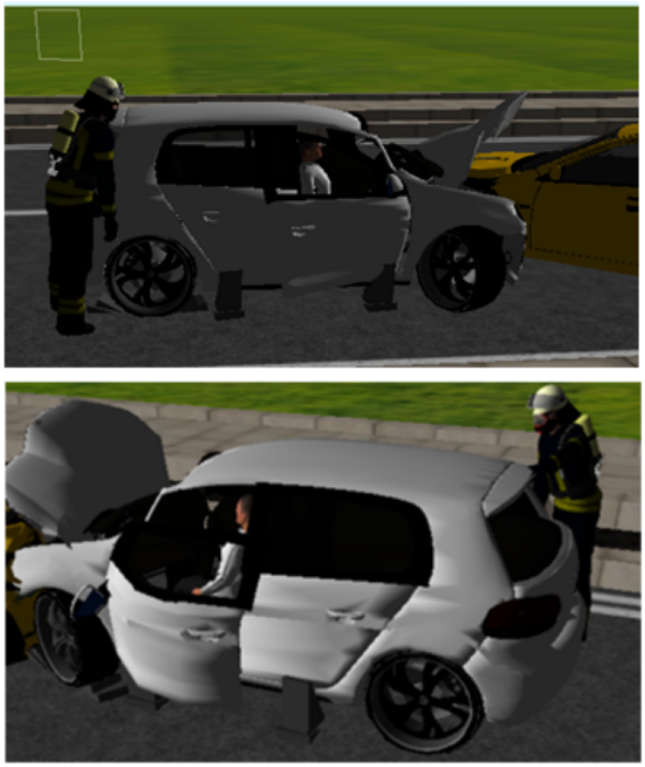

Figure 3. Tool operator stabilizing the vehicle.

fighter leader assigns the tool operator squad to stable the vehicle. Two tool operators navigate to the tool staging area to pick up the stable tools, and move to the sills and wheels respectively. Table 1 presents the stable status from both tool operators. Tool operator 1 starts to stable the sills and tool operator 2 begins with the wheels

	stabled sills	stabled wheels
tool operator 1	3	0
tool operator 2	1	2
tool operator squad	4	2

Table 1. Vehicle stabilization result.

stabilization. After tool operator 2 finishing both wheels, tool operator 1 is still working on the third sill, so that there is one sill remain and tool operator 2 decides to help tool operator 1 to stable the last sill. After all sills and wheels are stabilized, they move back to the spare location, and report to the firefighter leader the fulfillment of the task and waiting to perform further tasks. The firefighter leader considers the StabilizaVehicle task as fulfilled and assigns the further tasks, e.g. SpaceCreate, to the squads.

4.2 Ease of knowledge representation

HTN approaches are appropriate in describing the relations between actions [5]. As figure 1 presented, all decomposed methods and tasks are presented to accomplish the compound task `StabilizeVehicle` which matches the firefighter hierarchy, so the task network is clearly resented to each role or each level in the hierarchical structured firefighter domain. Moreover, actions and relations are expressive, e.g. the first method of `StableSills` allows the firefighter in tool operator squad to stable the sills using appropriate stable tools: the tool operator will find the correct tool in tool stage area or ask for backup staff to get the tool; and then go to the unstable sill; when reaches the sill, a stabilization action will be performed. Therefore, knowledge is intuitively represented via HTN.

5 CONCLUSION AND FUTURE WORKS

Although serious games solve the time-consuming and expensive problem to some extent if compared with real life firefighter training, it is still a problem to train firefighters in RTC scenario without attendance of all involved group members or customized individual training without autonomous NPCs. This paper introduces a HTN-based AI system which took the “hierarchical” advantage of HTN planning technique to substitute real firefighter members with believable NPCs to accomplish tasks based on the study of the build-in hierarchical structured firefighter domain. We take the task StabilizeVehicel as an example to represent issuing hierarchical orders in the firefighter domain, and create an according task sequence to fulfill the task via two squad members from tool operator squad. As presented in Chapter 3.2, orders from firefighter domain have been easily translated via methods, which recursively decompose the compound tasks to primitive ones so that the firefighter squad members can fulfill them. Additionally, how methods decomposing compound tasks depends on the world states, e.g. stabilize vehicle depends on how the crashed car resting, which will affect how the operator squad perform the task and which tools to chosen. The NPCs will make the decisions to fulfill tasks according to the world state. Therefore, the system is capable of create autonomous NPCs that can follow a standard firefighter guideline to perform goal tasks in a hierarchical task plan network according to the on going world states.

The HTN-based AI system works efficiently to handle the task vehicle stabilization among believable NPCs. However, to stable the

vehicle is only one of many tasks to rescue the trapped casualty. In the future, a complete scene will be developed to test the more complected interactions among NPCs, such as cooperation and communication rather than only issuing orders.

REFERENCES

[1] *Fire and Rescue Service manual: Volume 2: Fire service operations, Incidents involving rescue from road vehicles*, The Stationery Office, HM Fire Service Inspectorate Publications Section, 2007.

[2] *Fire Service Manual: Incident command, Volume 2*, The Stationery Office, HM Fire Service Inspectorate Publications Section, 2008.

[3] *Technisch-medizinischen Rettung nach Verkehrsunfaellen*, Technisch-Wissenschaftlicher Beirat (TWB), der Vereinigung zur Foerderung des Deutschen Brandschutzes e.V., 2009.

[4] C. Bohlens and P.G. Knacke, *Fallbeispiele Rettungsdienst 2*, Fallbeispiele Rettungsdienst, Stumpf und Kossendey, 2006.

[5] Kutluhan Erol, James Hendler, and Dana S Nau, 'Htn planning: Complexity and expressivity'.

[6] Kenneth Hullett and Michael Mateas, 'Scenario generation for emergency rescue training games', in *Proceedings of the 4th International Conference on Foundations of Digital Games*, pp. 99–106. ACM, (2009).

[7] Troy Humphreys, 'Exploring htn planners through examples', *Game AI Pro: Collected Wisdom of Game AI Professionals*, 149, (2013).

[8] Hans Kemper, *Führen und leiten im Einsatz*, Hüthig Jehle Rehm, 2008.

[9] Ibrahim M Mahmoud, Lianchao Li, Dieter Wloka, and Mostafa Z Ali, 'Believable npcs in serious games: Htn planning approach based on visual perception', in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pp. 1–8. IEEE, (2014).

[10] Debajyoti Mukhopadhyay, Falguni Chathly, and Nagesh Jadhav, 'Business oriented web service processing and discovery framework', in *Advances in Computing, Communication, and Control*, 156–166, Springer, (2013).

[11] Rescuesim.com. Rescuesim - incident command simulator, 2016.

[12] Gheorghe Tecuci, Mihai Boicu, Thomas Hajduk, Dorin Marcu, Marcel Barbulescu, Cristina Boicu, and Vu Le, 'A tool for training and assistance in emergency response planning', in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pp. 23–23. IEEE, (2007).

[13] Xvrsim.com. Virtual reality training software for safety and security, 2016.