

## A Tool for Training and Assistance in Emergency Response Planning

Gheorghe Tecuci, Mihai Boicu, Thomas Hajduk,

Dorin Marcu, Marcel Barbulescu, Cristina Boicu, Vu Le

*Learning Agents Center, George Mason University, 4400 University Dr., Fairfax, VA 22030*

*{tecuci, mboicu, dmarcu, mbarbule, ccascava, vle3}@gmu.edu, THajduk@aol.com*

### Abstract

*This paper presents research on developing a new type of software tool for training and assisting the personnel in emergency response planning. The tool, called Disciple-VPT, will include a library of virtual planning experts, each with a certain level of expertise (such as basic, intermediate or advanced) in one of the 15 emergency support functions defined by the US National Response Plan. Disciple-VPT can be used in a variety of training exercises where responders learn from virtual experts how to collaborate in emergency response planning. The development of Disciple-VPT is facilitated by the Disciple-VE learning agent shell that can be taught by a subject matter expert how to plan, through examples and explanations, in a way that is similar to how the expert would teach an apprentice.*

### 1. Introduction

This paper presents a new type of software tool, called Disciple-VPT (Virtual Planning Team), to train and assist the personnel in emergency response planning.

In the context of this paper, *planning means finding a partially ordered set of elementary actions that perform a complex task* [1].

Let us consider the task of planning the response to an emergency situation, such as a natural disaster or a terrorist attack. The US National Response Plan [2] identifies 15 primary emergency support functions performed by federal agencies in emergency situations. Each such function defines an expertise domain, such as emergency management; police operations; fire department operations; hazardous materials handling; health and emergency medical services; sheltering, public works and facilities; and federal law enforcement. People with expertise in these areas have to collaborate to plan the response to emergency situations. Similarly local and state agencies undertake these functions responding to such emergencies without or before any federal assistance is provided.

A sample emergency situation is the following one. “While waiting for a light to change at the intersection of Braddock Road and Ox Road, the driver of a tanker truck is told by a passer that there is a liquid pouring from underneath the truck’s trailer. When the light changes the truck turns into the George Mason University (Mason) campus looking for a place to park. A Mason police officer observes the truck and stops it on Patriot Circle near Liberty Square. The truck is a MC-312 corrosive liquid tanker trailer carrying red fuming nitric acid. The weather conditions are light rain with a temperature of 77 degrees F. The forecast is for continued light rain.”

Planning the appropriate response to this situation requires the involvement of 8 experts in the areas mentioned above. The generated plan has to include actions for the establishment and use of an Incident Command System, for dealing with a hazardous and toxic substance, for the establishment and evacuation of hot and warm zones. It also has to include life safety measures, such as evacuation, decontamination and treatment, as well as offensive and defensive control measures.

The problem that we are investigating is how to train the personnel to collaborate in planning the response to such situations. Disciple-VPT is developed as a general software tool that will support this training. It includes a library of virtual experts and a capability for creating multi-agent planning systems for a variety of emergency situations.

Each *virtual expert (VE)* is a knowledge-based agent that can rapidly acquire the planning expertise of a subject matter expert, and can collaborate with other virtual experts to develop plans that are beyond the capabilities of individual virtual experts.

Our goal is to develop a library of virtual experts, each virtual expert having the planning knowledge that corresponds to one of the 15 primary emergency support functions mentioned above, and to a certain level of expertise in that function. For instance, a virtual expert may represent a *basic* emergency management expert or an *advanced* fire department

operations expert (see Section 5 for more details on these agents).

Disciple-VPT allows the specification of an emergency scenario and the formation of a team of virtual experts (extracted from its library) that will collaborate in generating a plan for responding to that situation. Emergency personnel can learn from the planning performed by the virtual experts, can collaborate with complementary virtual experts, and can also use the virtual experts as assistants in planning the actual response to a given situation.

The rest of this paper is organized as follows. The next section presents an overview of Disciple-VPT from the end-user's perspective. Section 3 presents the emergency scenario (a propane tank on fire) that was used to guide the research, development and evaluation of Disciple-VPT. Examples of planning tasks from this scenario will help present the features of Disciple-VPT. Section 4 presents the architecture and use of the Disciple-VE learning agent shell which is at the basis of the capabilities of Disciple-VPT. Section 5 discusses the development of two virtual experts, one for fire operations and the other for emergency management. Section 6 presents the evaluation of the current version of Disciple-VPT. Finally, section 7 summarizes the main research contributions and the future research and development directions.

## 2. Overview of Disciple-VPT

Figure 1 presents the end-user's view of the three major components of Disciple-VPT:

- **VE Assistant**, an agent that supports the user in using Disciple-VPT;
- **VE Library**, an extensible library of virtual planning experts;
- **VE Team**, a dynamically assembled team of virtual experts selected from the VE Library.

### 2.1. The VE Assistant

The user of Disciple-VPT is guided by the **VE Assistant** to specify an emergency situation and the profiles of several responders that have to collaborate in planning the response to that situation. Then the VE Assistant identifies the virtual experts from the VE Library that most closely match the specified profiles of the responders, and forms the VE Team. This VE Team then simulates the planning performed by the responders. The VE Assistant also helps the user to browse, understand and compare the various plans generated by the VE team.

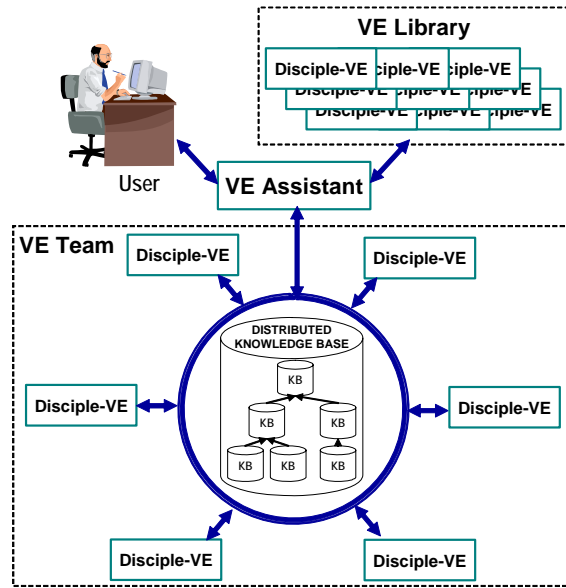


Figure 1. Use of Disciple-VPT

### 2.2. The VE Library

The **VE Library** is an extensible library of virtual planning experts. Each virtual planning expert models the planning expertise of a human expert, characterized by a certain expertise domain and a certain level of expertise in that domain. The knowledge bases of these virtual experts are hierarchically organized, as illustrated in Figure 2.

In this illustration there are four expertise domains, D1, D2, D3, and D4, and nine virtual experts, each associated with a knowledge base from the bottom of the hierarchy. Each virtual expert engine VE is a customization of the Disciple-VE shell which will be described in Section 4. The three left-most virtual experts are experts in the domain D1 with different levels of expertise: basic, intermediary and advanced. In addition to their specific knowledge base (e.g. KB-B1), they all inherit general knowledge about the domain D1, knowledge represented in KB-D1. They also inherit knowledge from the higher level

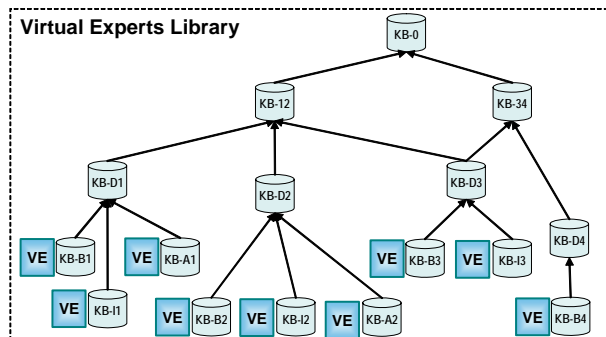


Figure 2. The organization of the VE Library

knowledge bases KB-12 and KB-0. These higher level knowledge bases contain general knowledge, useful to many agents, such as an ontology of units of measure, a time ontology, or a space ontology.

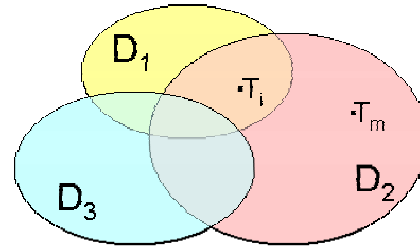
Traditional knowledge engineering practice builds each knowledge base from scratch, with no knowledge reuse, despite the fact that this is a very time-consuming, difficult and error-prone process [3, 4, 5]. On the contrary, the hierarchy of knowledge bases from the VE Library offers a practical solution to the knowledge reuse problem, speeding up the process of building a new virtual expert. Consider, for example, developing a new virtual expert for the D3 domain. This expert will already start with a KB composed of KB-D3, KB-12, KB-34, and KB-0. Thus, the VE Library can also be regarded as an existing knowledge repository for the new virtual experts to be developed.

### 2.3. The VE Team

The *VE Team* is a team of virtual experts selected from the VE Library. Each virtual expert, referred to as Disciple-VE, has the same Disciple-based architecture (see Section 4) but a different knowledge base. These virtual experts (e.g. an emergency management expert, an expert in hazardous materials handling, a federal law enforcement expert) collaborate in plan generation.

Although each virtual agent is expert in a certain expertise domain, these expertise domains are not disjoint, but overlapping, as illustrated in Figure 3. In this illustration the planning task  $T_m$  belongs only to  $D_2$  and can only be performed by a virtual expert from that domain.  $T_i$  is a task common to  $D_1$  and  $D_2$  and can, in principle, be performed either by a virtual expert in  $D_1$  or by a virtual expert in  $D_2$ . In general, a virtual expert will only cover a part of a given expertise domain, depending on its level of expertise.

A virtual expert has partial knowledge about its ability to generate plans for a given task, knowledge that is improved through learning. For instance, the virtual expert knows that it may be able to generate plans for a given task instantiation because that task belongs to its expertise domain, or because it was able to solve other instantiations of that task in the past. Similarly, it knows when a task does not belong to its area of expertise. The virtual experts, however, do not require predefined knowledge about the problem solving capabilities of the other experts from a VE Team, or from the VE Library. This is a very important feature of Disciple-VPT that facilitates the addition of new agents to the library, or the improvement of the existing agents, because this will not require taking into account the knowledge of the other agents.



**Figure 3. Sample coverage of expertise domains**

The multi-domain approach to expert problem solving represented by Disciple-VPT extends significantly the applicability of the traditional expert systems [3, 4, 5]. These expert systems are limited to a single expertise domain and their performance decreases dramatically when attempting to solve problems that have elements outside their narrow domain of expertise. On the contrary, a Disciple-VPT type system can efficiently solve such problems by incorporating additional virtual experts. Because many of the expert tasks actually require collaboration with other experts, one can claim that the type of system represented by Disciple-VPT is more adequate for solving real-world problems.

### 2.4. Use of Disciple-VPT

There are many ways in which a fully-functional Disciple-VPT system can be used for training or actual planning assistance during an emergency. For instance, it can be used to develop a wide range of training scenarios by guiding the user to select between different scenario characteristics.

Disciple-VPT can also be used to assemble teams of virtual planning experts that can demonstrate and teach how people should plan the response to various emergency situations. Another approach is to assemble combined teams which include both people and virtual experts. The team members will then collaborate in planning the response to the generated emergency scenario. In a combined team, human responders can play certain emergency support functions by themselves, or can play these functions with the assistance of corresponding virtual experts. During the training exercise a responder who has a certain emergency support function will learn how to perform that function from a corresponding virtual expert with higher competence. The responder will also learn how to collaborate with the other responders or virtual experts that perform complementary support functions.

The next section introduces the emergency scenario that was used to guide the research, development and evaluation of Disciple-VPT.

### 3. Sample Emergency Scenario

A sample emergency scenario was presented in Section 1. It involved a tanker trailer leaking red fuming nitric acid near a residential area. In the rest of this paper we will use another scenario to describe our approach to rapid development of virtual experts for multi-domain collaborative planning.

“Workers at the propane bulk storage facility in Gainesville, Virginia, have been transferring propane from a train car to fill one of two 30,000 gallon bulk storage tanks. A fire is discovered in the fill pipe at the bulk tank and a large fire is developing. The time is 15:12 on a Wednesday in the month of May. The temperature is 72 degrees and there is a light breeze out of the west. The roads are dry and traffic volume is moderate. The fire department is summoned to the scene 5 minutes after the fire started. The facility is located in a rapidly growing area 2,000 ft from an interstate highway and 200 ft from two heavily traveled US highways. New shopping centers have popped up in the area including food stores, large box building supply facilities, and large box retail facilities. These facilities are accompanied by fast food restaurants and smaller retail stores. Residential concentrations include approximately 2400 residents. The Local Emergency Operations Plan has all the required components including Public Information, Communications, and Sheltering. Shelters utilize schools managed by the Red Cross. The Virginia Department of Transportation provides highway services.”

Planning the appropriate response to this emergency situation requires the collaboration of experts in fire department operations, in emergency management, and in police operations. The generated plan will consist of hundreds of partially order actions. One group of actions deal with the arrival of resources such as fire units, emergency management services units, police units, as well as individuals with different areas of expertise (e.g. emergency manager, safety officer, highway supervisor, planning officer, training logistics officer, public information officer).

Another group of actions deal with the establishment of the structure of the Incident Command System (ICS) and the allocation of resources based on the evaluation of the situation. The structure of the ICS follows the standard National Incident Management System [6]. The National Incident Management System was approved March 1, 2004 and establishes standard incident management processes, protocols and procedures so that all local, state, federal and private-sector emergency responders can coordinate their responses, share a common focus

and more effectively resolve events. Its main components are the unified command, the command staff, and the general staff. The structure and organization of these components depend on the current situation. For example, in the case of the above scenario, the unified command includes representatives from the fire department, police department, highway department, and propane company. The command staff includes a safety officer, a public information officer and a liaison officer. The general staff includes an operation section, a planning section, a logistics section, and a finance and administration section. Each of these sections is further structured and staffed.

Yet other groups of actions deal with the various activities performed by the components of the Incident Command System. For instance, the fire management group performs the cooling of the propane tank with water. The evacuation branch evacuates the Gainesville hot zone. The emergency manager arranges for transportation, sheltering, and emergency announcements to support the evacuation. The perimeter control branch implements the perimeter control for the Gainesville hot zone. The traffic control branch implements the traffic control to facilitate the evacuation of the Gainesville hot zone. The Gainesville command establishes rapid intervention task forces to respond if the propane tank explodes.

One difficulty in generating such a plan, apart from the fact that it involves many actions, actors, and resources, is that the actions from the above groups are actually performed in parallel. Our goal is to create a capability for rapid and low cost development of virtual planning experts to be used in this type of multi-domain collaborative planning. Moreover, the plans generated by the system should be more comprehensive than those produced by a collaborative team of humans, and should be generated much faster and cheaper than currently possible. The next section introduces the Disciple-VE learning agent shell which is at the basis of creating the above capability.

### 4. The Disciple-VE Learning Agent Shell

Critical to the generality and usefulness of Disciple-VPT is the ability to rapidly develop virtual experts that have the expertise of specific human experts.

For many years we have researched *a theory, methodology and family of tools for the development of knowledge-based agents by subject matter experts, with limited assistance from knowledge engineers, called the Disciple approach* [8, 9, 10].

The main goal of the Disciple approach is to overcome the knowledge acquisition bottleneck in the development of knowledge-based systems [4]. Its basic

idea is to develop a general problem solving and learning agent which has no specific knowledge in its knowledge base, but can be taught by a subject matter expert how to solve problems, through problem solving examples and explanations, in a way that is similar to how the expert would teach an apprentice. We call such an agent, a Disciple learning agent shell.

Previous versions of the Disciple learning agent shell were used to develop agents for course of action critiquing and center of gravity analysis, and were successfully evaluated as part of the DARPA's High Performance Knowledge Bases and Rapid Knowledge Formation programs [9]. The Disciple agents for center of gravity analysis are used in several courses at the US Army War College since 2001 [10]. In the "Case Studies in Center of Gravity Analysis" course the students (who are high level officers from all the military services) use trained Disciple agents as intelligent assistants that help them develop a center of gravity analysis of a war scenario. In the follow-on course, "Military Applications of Artificial Intelligence" the students, now experts in center of gravity analysis, teach personal Disciple agents their own expertise, and then evaluate both the developed agents and the development process.

Disciple-VPT incorporates the Disciple-VE learning agent shell which extends significantly the previous versions of the Disciple shells with capabilities of learning for planning and of rapid development of knowledge-based planners.

The general planning paradigm of Disciple-VE is **Hierarchical Task Network (HTN) planning** the goal of which is to find a partially ordered set of elementary actions that perform a complex task in a given world state, by successively reducing the task to simpler and simpler tasks, down to the level of elementary actions [1, 7]. HTN planning is the planning approach that has been used for practical applications more than any other approach because it is closer to how human experts think when solving a planning problem [1].

Figure 4 shows the top level reduction step for the task "Respond to the Gainsville incident" described in Section 3. This task is reduced to 5 sub-tasks, meaning that it can be performed by performing its 5 sub-tasks. A **task** is characterized by name, preconditions and goal. In order for a task to be considered for execution in a given state of the world, its preconditions have to be satisfied. Successful execution of the task will lead to a new state of the world in which the goal is achieved. The sub-tasks of a task are partially ordered. For example Sub-task1, Sub-task2, and Sub-task3 can be performed in parallel, but Sub-task 4 has to be performed after Sub-task 3. A task reduction step may be guided by a question/answer pair.

A fragment of the plan of actions for performing the task in Figure 4 is presented in Table 1. An **action** is characterized by name, preconditions, delete effects, add effects, resources and duration. An action can be performed in a given state of the world  $S_i$  if the action's preconditions are satisfied in that state. Action's execution has duration and requires the use of some resources. The resources are objects from the state  $S_i$  that are uniquely used by this action during its execution. This means that any other action that would need some of these resources cannot be executed in parallel with it. As a result of action's execution the state  $S_i$  changes into the state  $S_j$ , as specified by the action's effects. The delete effects indicate what facts from the initial state  $S_i$  are no longer true in the final state  $S_j$ . The add effects indicate what new facts are true in final state  $S_j$ .

In order to perform HTN planning, the knowledge base of a Disciple-VE agent contains two main types of knowledge: an object ontology and reasoning rules.

The **object ontology** represents the objects and the types of objects from an application domain, together with their properties and relationships [11]. A fragment of the object ontology from the emergency planning domain is shown in Figure 5. For example, the "Gainsville incident" node from the bottom left side of Figure 5 represents the major fire emergency described in Section 3. As indicated in Figure 5, "Gainsville incident is\_caused\_by fire1" and "fire1 is\_fuelled\_by gas propane f1." A **world state** is represented by all the objects present in the world together with their properties and relationships at a given moment of time. For instance, the bottom part of Figure 5 shows the representation of part of the state of the world where fire1, which is situated in fill pipe1, is impinging on propane tank1.

A subject matter expert teaches Disciple-VE how to plan the performance of a complex task in a way that is similar to how the expert would teach a person. For instance, the expert may show the agent how to plan the performance of "Respond to Gainsville incident" (see Figure 4) and may help it to understand the reasoning process. The expert follows a task reduction paradigm where the initial task is successively reduced to simpler and simpler tasks, down to the level of elementary actions, as was illustrated in Figure 4. The partially ordered set of these elementary actions represents the plan for performing the initial task. From such planning examples the agent learns general planning rules that allow it to plan the performance of similar tasks [12].

As the agent learns general rules and builds its knowledge base, the expert-agent interaction evolves from a teacher-student interaction toward an

Reasoning Hierarchy Reasoning Step

Default Viewer  Advanced Viewer

Task

Task: Respond to the [Gainsville incident](#)

Goal: [Gainsville abstract incident action plan](#) has been implemented

Question

What kind of incident is [Gainsville incident](#)?

Answer

[Gainsville incident](#) is a major fire emergency because it involves a large propane tank on fire

Sub-task (1)

Abstract: Resources to respond to a major fire emergency arrive at the scene of the incident

Precondition: [Gainsville incident](#) is a major fire emergency

Task: Resources to respond to a major fire emergency arrive at the scene of the [Gainsville incident](#)

Goal: Resources to respond to the [Gainsville incident](#) have arrived at the scene

Sub-task (2)

Abstract: Establish the command of the Incident Command System for a major fire emergency

Precondition: [Gainsville incident](#) is a major fire emergency

Task: Establish the command of the Incident Command System for the [Gainsville incident](#)

Goal: [Gainsville command](#) is the command of the [Gainsville ICS](#)

After:  1

Sub-task (3)

Abstract: Command evaluates the incident and determines the [abstract incident action plan](#)

Precondition: [Gainsville command](#) is the command of the [Gainsville ICS](#), the incident command system for the [Gainsville incident](#)

Task: [Gainsville command](#) evaluates the [Gainsville incident](#) and determines the needed incident action plan.

Goal: [Gainsville abstract incident action plan](#) is the action plan for the [Gainsville incident](#)

After:  1  2

Sub-task (4)

Abstract: Command establishes the overall structure of the Incident Command System for a major fire emergency and allocates the resources based on the [abstract incident action plan](#)

Precondition: [Gainsville abstract incident action plan](#) is the action plan for the [Gainsville incident](#)

Task: [Gainsville command](#) establishes the overall structure of [Gainsville ICS](#) and allocates the resources based on the [Gainsville abstract incident action plan](#)

Goal: The structure of the [Gainsville ICS](#) has been established

After:  1  2  3

Sub-task (5)

Abstract: Incident Command System implements the [abstract incident action plan](#) for the incident

Precondition: [Gainsville ICS](#) and [Gainsville abstract incident action plan](#) are the incident command system and the abstract incident action plan for the [Gainsville incident](#)

Task: [Gainsville ICS](#) implements the [Gainsville abstract incident action plan](#) for the [Gainsville incident](#)

Goal: [Gainsville abstract incident action plan](#) has been implemented

After:  1  2  3  4

Q/A  Explanations  Ontology Specs  Abstract  Resources  Durations

Figure 4. Top-level task reduction step shown in the Reasoning Step Viewer



interaction where both collaborate in planning. During this joint planning process, the agent learns not only from the contributions of the expert, but also from its own successful or unsuccessful planning attempts.

The main strength of the Disciple agents comes from the fact that they can easily and rapidly learn such rules from subject matter experts.

The Disciple-VE shell is used to rapidly develop a Disciple-VE agent for a specific planning domain by following a two phase process:

- The development of an initial object ontology for that domain, which is performed jointly by a knowledge engineer and a subject matter expert, and
- The teaching of Disciple-VE, which is performed by the subject matter expert, with limited assistance from the knowledge engineer.

### 5. Basic Virtual Experts

We have developed two basic virtual experts, a fire department operations expert, and an emergency management expert. The development of these virtual experts was guided by the toxic substance leaking scenario described in Section 1 and by the propane tank on fire scenario described in Section 3.

First we have worked with a subject matter expert to model the plan generation process for these two

scenarios by using the task reduction paradigm, as illustrated in Figure 4.

Besides the development of the two reasoning trees, another result of this modeling process was the development of a modeling methodology which is presented in detail in [12]. Yet another result of this modeling process was the identification of the object concepts that need to be present in Disciple’s ontology so that it can perform this type of reasoning. Based on this specification of the object ontology, and using the ontology development tools of the Disciple-VE shell, we have developed an object ontology consisting of 410 concepts, 172 feature definitions, 319 generic instances, and 944 facts. A fragment of this ontology is presented in Figure 5.

Although we have worked with both scenarios mentioned above to develop the modeling trees and the ontology, further development of the fire expert and of the emergency management expert was only based on the propane tank on fire scenario. This development consisted in teaching the two virtual experts how to perform inference and to plan by using the propane tank on fire scenario. As a result of the teaching process, the virtual fire expert learned 81 planning rules, and the virtual emergency management expert learned 47 planning rules. The current versions of the

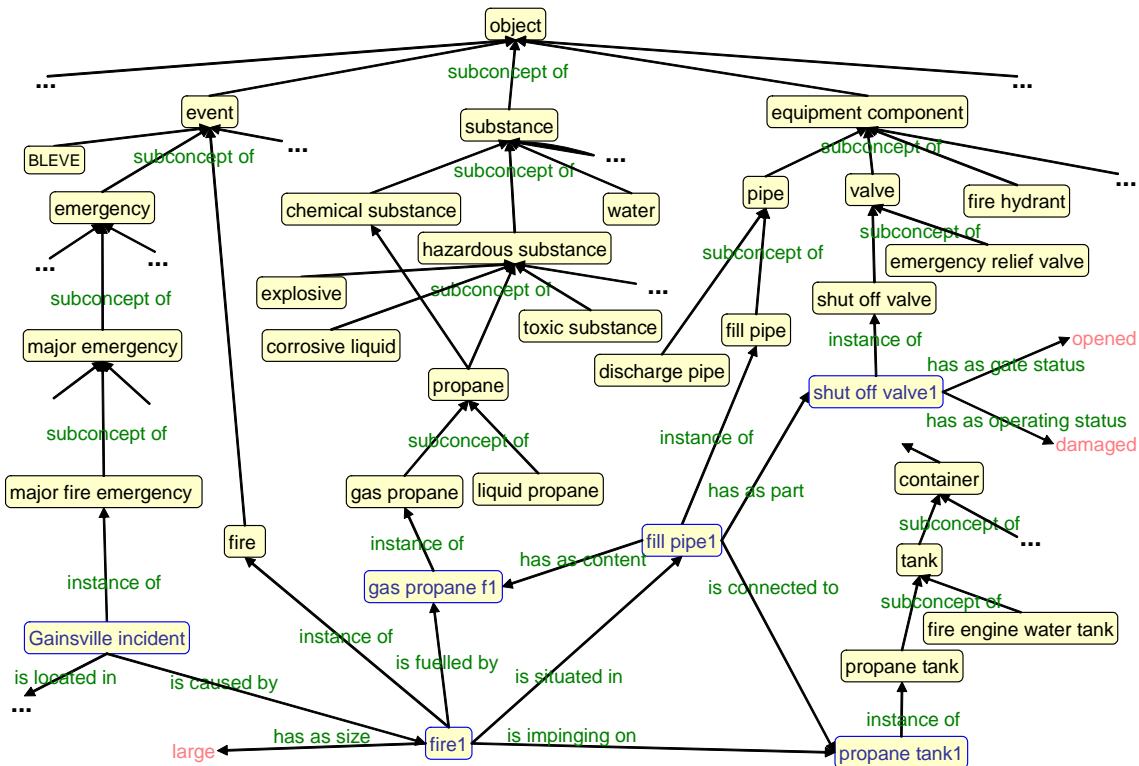


Figure 5. Fragment of the object ontology from the emergency planning area

Table 1. Fragment of the plan generated by Disciple-VPT

Id	Action	Result	Resources	Start Time	Duration
1	Suburbane Propane Company facility manager, a person, arrives at the scene of the Gainsville incident	<b>Add:</b> Suburbane Propane Company facility manager arrived at the scene and is <b>available</b> to take required actions	Suburbane Propane Company facility manager	0.0 s	1.0 min 0.0 s
2	ALS unit M504, an ALS unit, arrives at the scene of the Gainsville incident	<b>Add:</b> ALS unit M504 arrived at the scene and is <b>available</b> to take required actions	ALS unit M504, paramedic 504a, and paramedic 504b	0.0 s	5.0 min 0.0 s
3	fire engine company E504, a fire engine company, arrives at the scene of the Gainsville incident	<b>Add:</b> fire engine company E504 arrived at the scene and is <b>available</b> to take required actions	fire engine driver E504, fire fighter E504b, fire engine company E504, fire engine E504, deluge nozzle E504, water hose E504, fire officer E504, and fire fighter E504a	0.0 s	5.0 min 0.0 s
...	...	...	...	...	...
48	fire officer E504 assumes the command of the incident command system for the Gainsville incident, as fire department representative in the ICS unified command	<b>Delete:</b> fire officer E504 is no longer <b>available</b> <b>Add:</b> Gainsville ICS, the incident command system for the Gainsville incident, is created and fire officer E504 assumes ICS command as fire department representative in the Gainsville command	fire officer E504	5.0 min 0.0 s	15.0 s
49	Gainsville command evaluates the situation created by the Gainsville incident	<b>Add:</b> overpressure situation with danger of BLEVE in propane tank1 is caused by fire1	Gainsville command	5.0 min 15.0 s	30.0 s
50	Gainsville command determines the abstract incident action plan for overpressure situation	<b>Add:</b> The plan is to apply cooling water, to evacuate people from 1.0 mi around propane tank1, to perform traffic control, perimeter control, and to establish rapid intervention task forces	Gainsville command	5.0 min 45.0 s	30.0 s
...	...	...	...	...	...
54	fire engine company E504 sets up water hose E504 to apply water to propane tank1	<b>Delete:</b> water hose E504 is <b>available</b> <b>Add:</b> water hose E504 is assigned	fire engine company E504, fire engine driver E504, fire fighter E504b, fire fighter E504a, water hose E504, and fire engine E504	8.0 min 30.0 s	3.0 min 0.0 s
55	fire engine company E525 drops off deluge nozzle E525 for fire engine company E504	<b>Delete:</b> deluge nozzle E525 that belongs to fire engine E525 is no longer <b>available</b> <b>Add:</b> deluge nozzle E525 is <b>assigned</b>	fire fighter E525a, fire officer E525, deluge nozzle E525, fire engine driver E525, and fire fighter E525b	8.0 min 30.0 s	2.0 min 0.0 s
...	...	...	...	...	...
63	fire engine company E525 establishes continuous water supply from fire hydrant1 for fire engine company E504	<b>Add:</b> fire hydrant1 is assigned to fire engine company E504	fire fighter E525a, fire fighter E525b, fire officer E525, fire engine driver E525, and fire engine company E525	14.0 min 30.0 s	7.0 min 0.0 s
...	...	...	...	...	...



two developed virtual experts share the object ontology. This ontology will be split into shared and private ontologies in the next versions of these agents.

## 6. Evaluation

To evaluate the Disciple-VPT system we have developed a *scenario pattern* based on the scenario from Section 3. Then we have asked a fire department operations expert and an emergency management expert to define a specific scenario based on this pattern, by providing the missing elements. For example, the two experts decided on the day and time of the incident, the position of the fire with respect to the propane tank, the estimated amount of propane in the tank, the available resources (e.g. fire engine companies, truck engine companies, hazmat units, police personnel, county personnel, etc.) and when are they arriving at the scene of the incident. After that, both the team of human experts and Disciple-VPT generated plans to respond to this emergency situation.

The plan generated by Disciple-VPT consisted of 89 partially ordered elementary actions. A fragment of this plan is shown in Table 1. This plan was evaluated by the above mentioned experts and by the expert with whom we have developed the agents. Then each expert has filled-in a questionnaire. The questionnaire included statements about various characteristics of the generated plan and about the Disciple-VPT system. The experts were asked to indicate whether they strongly agree (SA), agree (A), are neutral (N), disagree (D) or strongly disagree (SD) with these statements.

The top part of Table 2 presents some of the results of the evaluation of the plan generated by Disciple-VPT, which was considered good and easy to understand by the human experts. One of the experts disagreed with the way Disciple-VPT ordered some of the actions. However, this reflected a disagreement between the evaluating experts and not a planning error, the generated order being that taught by the expert who instructed Disciple-VPT.

The evaluation of the hierarchical plan generation process (see the middle of Table 2) shows that it significantly improves the understandability of this process. Finally the experts agreed or strongly agreed that Disciple-VPT has many potential applications in emergency response planning, from developing exercises for training, to actual training, and to its use as planning assistant.

A limitation of this evaluation is that it is based on the opinion of only 3 experts. More experts are needed in order for the results to have statistical significance.

**Table 2. Evaluation Results**

Generated Plan	SA	A	N	D	SD
The generated plan is easy to understand.		3			
The generated plan consists of a good sequence of actions.		2		1	
The objectives of the actions are clear from the generated plan.		3			
The generated plan has a good management of the resources.	1	1	1		
The level of detail in the plan is appropriate.		3			

Hierarchical Plan Generation Process	SA	A	N	D	SD
The hierarchical task-reduction structure makes the logic of the plan clear.	2	1			
The hierarchical task-reduction structure makes the goals of the plan clear.	1	2			
The questions, the answers, and the preconditions help understand the logic of the plan generation process.	2	1			
The preconditions, effects, duration, and resources, make the specification of the plan's actions clear.	1	2			
The hierarchical task-reduction structure may be used to teach new persons how to plan.	3				

Usability of Disciple-VPT	SA	A	N	D	SD
Disciple-VPT could be used in developing and carrying out exercises for emergency response planning.	2	1			
Disciple-VPT could be used for training the personnel for emergency response planning.	2	1			
Disciple-VPT could be used as an assistant to typical users, guiding them how to respond to an emergency situation.		3			

## 7. Contributions and Future Directions

The research presented in this paper represents a major extension of the Disciple theory and methodology for the development of knowledge-based agents by subject matter experts, with limited assistance from knowledge engineers.

First we have extended the Disciple approach to allow the development of complex HTN planning agents that can be taught their planning knowledge, rather than having it defined by a knowledge engineer. This is a new and very powerful capability that is not present in current action planning systems [1, 7]. This capability was made possible by several major developments of the Disciple approach. For instance,

we have significantly extended the knowledge representation and management of a Disciple agent by introducing new types of knowledge that are characteristic to planning systems, such as planning tasks and actions (with preconditions, effects, goal, duration, and resources) and new types of rules (e.g. planning tasks reduction rules, concretion rules, action rules, goal synthesis rules). We have introduced state knowledge bases and have developed the ability to manage the evolution of the states in planning. We have developed a modeling language and a set of guidelines that help subject matter experts express their planning process. We have developed an integrated set of learning methods for planning, allowing the agent to learn general planning knowledge from a single planning example formulated by the expert.

A second accomplishment is the development of an integrated approach to planning and inference, both processes being based on the task reduction paradigm. This improves the power of the planning systems that can now include complex inference trees. It also improves the efficiency of the planning process because some of the planning operations can be performed as part of a much more efficient inference process that does not require a simulation of the change of the state of the world.

A third accomplishment is the development and implementation of the concept of library of virtual experts. This required the development of methods and tools for the management of a hierarchical knowledge repository. The hierarchical organization of the knowledge bases of the virtual experts also serves as a knowledge repository that speeds-up the development of new virtual experts that can reuse the knowledge bases from the upper levels of this hierarchy.

A fourth accomplishment is the development of the multi-domain architecture of Disciple-VPT which extends the applicability of the current expert systems to problems whose solutions require knowledge of more than one domain, as discussed in Section 2.3.

A fifth accomplishment is the development of two basic virtual experts, a basic fire expert and a basic emergency management expert, that can collaborate to develop plans of actions that are beyond their individual capabilities.

Finally, a sixth accomplishment is the development of an approach and system that has high potential for supporting a wide range of emergency training and planning activities, as discussed in Section 2.4.

Future research will involve the development of additional agents for the VE Library, and of the Disciple-VPT system for actual use in training personnel for emergency response planning.

## 8. Acknowledgements

The authors are grateful to Susan Durham for inspiring, coordinating and supporting this research.

## 9. References

- [1] Ghallab M., Nau D., and Traverso P., *Automatic Planning: Theory and Practice*, Morgan Kaufmann, 2004.
- [2] *Department of Homeland Security, National Response Plan*, 2004, [www.dhs.gov/dhspublic/interapp/editorial/editorial\\_0566.xml](http://www.dhs.gov/dhspublic/interapp/editorial/editorial_0566.xml) (accessed on June 14, 2006).
- [3] Awad E., and Ghaziri M. H., *Knowledge Management*, Pearson Education Inc., Upper Saddle River, NJ, 2004.
- [4] Buchanan, B.G., and Wilkins, D.C. (eds.), *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems*, Morgan Kaufmann, San Mateo, CA, 1993.
- [5] Jackson P., *Introduction to Expert Systems*, Addison-Wesley, Essex, England, 1999.
- [6] *Federal Emergency Management Agency, National Incident Management System*, March 1, 2004, <http://www.fema.gov/emergency/nims/index.shtml> (accessed on June 14, 2006).
- [7] Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J., Wu, D., Yaman, F., SHOP2: An HTN planning system, *Journal of Artificial Intelligence Research*, 20, pp. 379–404, 2003.
- [8] Tecuci, G., *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*, London, England: Academic Press, 1998.
- [9] Tecuci G., Boicu M., Bowman M., and Marcu D., with a commentary by Burke M., An Innovative Application from the DARPA Knowledge Bases Programs: Rapid Development of a High Performance Knowledge Base for Course of Action Critiquing, *AI Magazine*, 22, 2:43-61, AAAI Press, Menlo Park, CA, 2001.
- [10] Tecuci G., Boicu M., Marcu D., Stanescu B., Boicu C., Comello J., Training and Using Disciple Agents: A Case Study in the Military Center of Gravity Analysis Domain, *AI Magazine*, 24, 4:51-68, AAAI Press, Menlo Park CA, 2002.
- [11] Fensel D., *Ontologies: A Silver Bullet for Knowledge Management*, Springer-Verlag, Berlin, 2000.
- [12] Tecuci G., Boicu M., Hajduk T., Marcu D., Barbulescu M., Boicu C., Le V., An Approach to Rapid Development of Virtual Experts for Multi-Domain Collaborative Planning, *Research Report*, Learning Agents Center, George Mason University, 2005.