

# An Attitude Based Multi-agent Problem Solving in a Hostile World

Madhu Goyal  
University of Technology, Sydney  
PO BOX 123  
Broadway NSW 2007 Australia  
madhu@it.uts.edu.au

*Abstract: In multi-agent setting agent often encounter conflicts in agents' plans and actions. This paper presents an attitude based cooperative decision making methodology that allows agents to act appropriately to various options in a hostile and dynamic fire world. It shows that attitude based decision making explore the attitudes and behaviors that help agents to solve problems constructively. The application and implementation of this methodology to a virtual fire- fighting domain has revealed a promising prospect in negotiating conflicts and solving them.*

*Keywords: teamwork, multi-agent systems, attitudes*

## 1. Introduction

The agents are entities within an environment, and they can sense and act [13]. This means that agents are not isolated entities and that they are able to communicate and collaborate with other entities. Simply put, agents that are not able to work together with other agents are destined to become virtually useless. Once agents are ready for collaboration, they will need to find the other agents they need to collaborate with. Such a task is easy if they know exactly which agents to contact and at which location. However, a static distribution of agents is very unlikely to exist: people are usually on the move and they are not always readily available to interact with others. The same holds true for dynamic multi-agent systems: agents need support to find other agents.

Collaborative work involves a group of autonomous agents choose to work together to achieve a common goal. One of the main questions in the multiagent systems community is how and why autonomous agents should cooperate with one another. Several formal and computational models of cooperative work or teamwork are currently developed and used within multi-agent systems research. One of the most popular model of teamwork is based on *joint intentions* [5]. This model essentially traces back to Toumella and Miller's [14] analysis of collective intentions. According to their view, collective intentions of a given set of agents, is assumed summation of intentions and beliefs of the members of the set. According to them, joint action by a team does not consist merely of simultaneous and coordinated individual actions, but a team must be aware of the status of the group effort as a whole. In this paper it is proposed that agents inhabiting complex multiagent dynamic worlds, while solving problems must be holding appropriate attitudes towards their physical and mental activities. This is particularly important when agents collectively engage in activities that span over prolonged periods of time. In this paper, it is argued that the notion of commitments as used in AI is too weak for agents to hold while solving problems in dynamic worlds particularly over prolonged periods of time and propose that agents need to hold appropriate attitudes towards the world objects and mental objects while solving problems. In this paper, a mental construct called *attitude* is proposed and its

significance in decision making in a dynamic fire world is discussed. The aim is to design teams, which can survive in a dynamic (virtual) world and solve problems with other similar agents inhabiting the same world. These agents will be capable of interacting with each other both individually and collectively. In addition, they will be capable of deriving plans and negotiating over the sharing of resources in order to achieve a common goal. The fire-fighting domain provides us with an excellent opportunity to design agents, which are capable of reactive as well as deliberative behavior.

## 2. Fire World Domain

The team ideas have been implemented on a simulation of fire world *FFTEAMS* using a virtual research campus. The idea of simulated fire world was first given in Phoenix [4], which is a real time, adaptive planner that manages forest fires in simulated environment. The virtual campus is implemented using C++ on Windows2000/NT platform, where more than 40 agents share the world via network. *FFTEAMS* is a dynamic, distributed, interactive, simulated fire environment where agents are working together to solve problems, for example, rescuing victims and extinguishing fire. The fire world *FFTEAMS* that we have considered in this paper consists of a large number of objects (of the order of hundreds) and several agents. It consists of several buildings, an open ground area, walkways, a car park, and campus gates. Objects in the fire world include walls, buildings, furniture, open areas and LPG gas tanks. There are two types of agents: victim agents and fire fighting agents. A part of our fire world is shown in figure 1. Our world is different from others' (like Air Combat [9] and RoboCup [12]) in respect that problems posed to the agents and the changes in the environment are not only caused by the actions of other agents but also by the changes the objects themselves undergo in the world (caused by the fire).

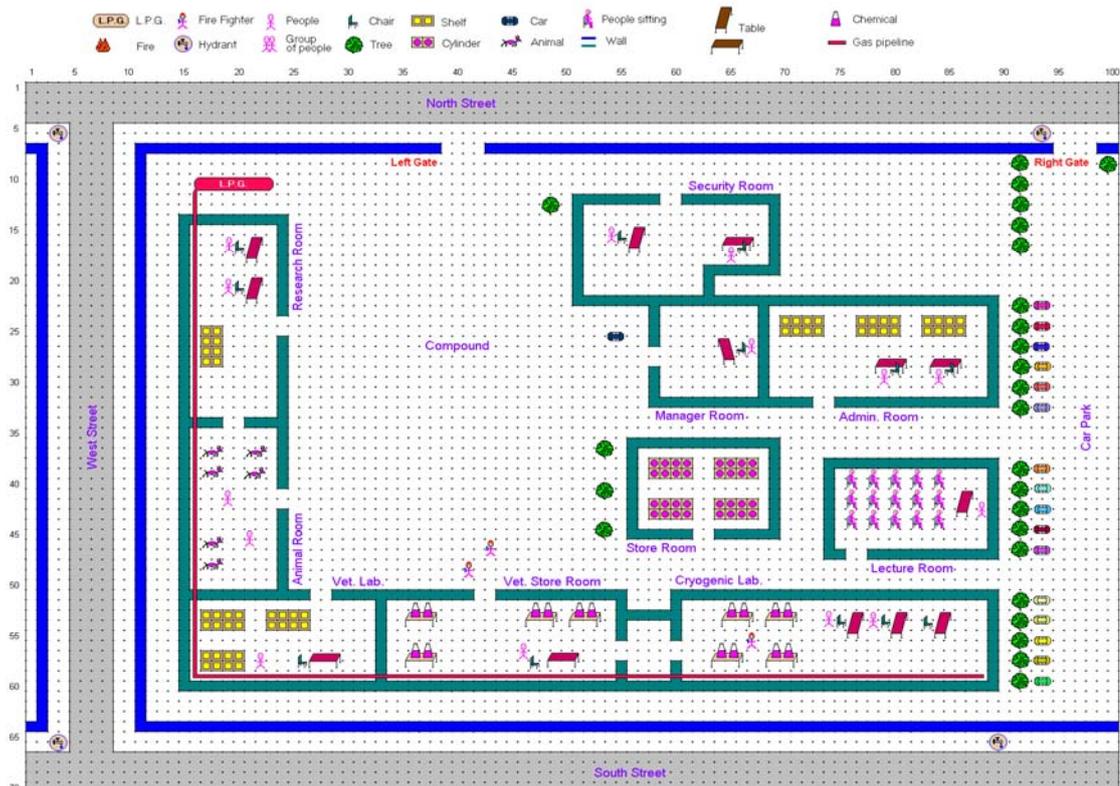


Figure 1: A Fire World

In a world such as this, no agent can have full knowledge of the whole world. Humans and animals in the fire world are modelled as autonomous and heterogeneous agents. While the animals run away from fire instinctively, the fire fighters can tackle and extinguish fire and the victims escape from fire in an intelligent fashion. An agent responds to fire at different levels. At the lower level, the agent burns like any object, such as chair. At the higher level, the agent reacts to fire by quickly performing actions, generating goals and achieving goals through plan execution. This world contains all the significant features of a dynamic environment and thus serves as a suitable domain for our team agents. Agents operating in the domain face a high level of uncertainty caused by the fire. Agents in the fire domain do not face the real time constraints as in other domains, where certain tasks have to be finished within the certain time. However, because of the hostile nature of the fire, there is strong motivation for an agent to complete a given goal as soon as possible. The communication and co-operation are the essentials to accomplish fire-fighting tasks and to achieve proper control of the hostile situation.

The team behaviours play an integral role in a Fire World. There are three main objectives for intelligent agents in the world during the event of fire: self-survival, saving objects including lives of animals and other agents and put-off fire. Because of the hostile settings of the domain, there exist a lot of challenging situations where agents do the team activities. Whenever there is fire, the basic team behaviour is exhibited by the fire fighters. The fire fighters perform all the tasks necessary to control an emergency scene. The problem solving activities of the fire fighters are putting out fire, rescuing victims and saving property. Apart from these primary activities there are a number of sub tasks eg. run towards the exit, move the objects out of the room, remove obstacles, and to prevent the spread of fire. In order to carry out the teamwork successfully the team of fire fighters should carry out number of procedures. There are number of preconditions of team behaviour in the fire world. *First*, the firefighters should work in pairs whenever they are in a hazardous or potentially hazardous location on the emergency scene. Firefighters working alone may over exert themselves or be unable to help themselves when trapped. *Second*, a fire attack must be coordinated to be successful. The fire fighters must perform the desired activity at the time when the officer wants them to perform. Depending on the conditions at the fire scene, the fire fighters may choose to perform immediate rescue or to protect exposures rather than attacking the fire. Coordination between team members performing different functions is crucial. For example, when carrying hose-lines the team of fire fighters requires an explicit coordination technique.

### **3. Multi-agent Problem Solving**

An agent intending to achieve a problem-solving goal must first commit itself to the goal by assigning the necessary resources, and then carry out the commitment when the appropriate opportunity comes. Carrying out the commitment typically involves planning for the goals, and executing the plans. The planning involves decomposing the complex goals, executing the plans and ultimately executing the primitive physical and communicative actions. The problem solving by group of agents is based on a set of mutually believed facts, commitments and conventions [5][15]. These researchers have presented an abstract model of cooperative problem solving, which describes all aspects of the process, from recognition of the potential for cooperation to team action. They have developed an explicit representation of joint goals and commitments, along with

an implicit (hardwired) response to commitment abandonment. In these cases, the basic idea is that responsibility for updating team members that a commitment has been abandoned resides with the agent that first drops the commitment. Typically, an agent that is dropping the commitment warns the others that it is doing so by explicitly sending them messages. But there are problems with this method eg. agent incapacitation and communication failure in stressed situations. This leads to view that it is not unreasonable to assign each agent the job of actively monitoring for itself the commitment of others to the joint goals. In practical terms, this means that agents should be continuously validating their commitments to the joint or team goal, and noticing when this commitment wavers [7].

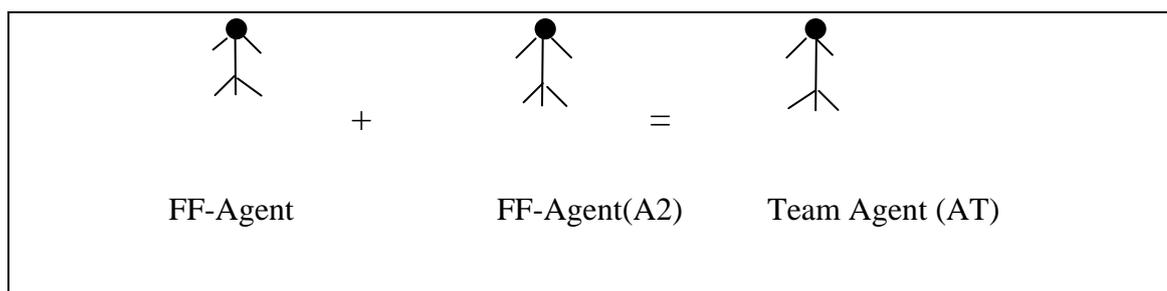


Figure 2: Fire-fighting Agents ( $A_i$ ) and Team Agents ( $A_T$ )

The attitude of agents specify the conditions under which the agents should reconsider its commitments and describes how the group of agents should behave both locally and with respect to its fellow group members if any such problem arises. The attitudes of agents provide the methods which clearly distinguishes between the situations, in which the commitment to the team action needs to be re-examined and the actions which should be taken in such circumstances. Thus the attitudes help agents to resolve conflicts arising due to individual goals, group goals, social goals and team goals and generate appropriate solutions to the local problems of the team agents with the help of various attitudes. In the fire world, problem solving involves the participation of victim agents and fire-fighting agents. The fire fighters in our world exist both as individual agents ( $A_i$ ) and as a team agent ( $A_T$ ) (Figure 2). The problem solving by abstract team agents ( $A_T$ ) is different from problem solving by the individual agents ( $A_i$ ) in the sense that problem solving by the team agent ( $A_T$ ) is a joint activity based on MB while the problem solving by individual agents is a joint activity dealing with the lower level execution issues or details. The team agent ( $A_T$ ) models *team* as a *collective* abstract attitude, while individual agent ( $A_i$ ) models *team* as an *individual* attitude towards the various attributes of team.

### 3.1 Attitude

Attitude is a *learned predisposition to respond in a consistently favourable or unfavourable manner with respect to a given object* [6]. So we define attitude as a *predisposition to respond consistently in favourable or unfavourable manner with respect to a given object*. In other words, the attitude is a preparation in advance of the actual response, constitutes an important determinant of the ensuing behaviour. However this definition seems too abstract for computational purposes. In AI, the fundamental notions to generate the desirable behaviours of the agents often include goals, beliefs, intentions, and commitments. Goal is a subset of states, and belief is a proposition that is held as true by an agent. Bratman [2] addresses the problem of defining the nature of intentions. Crucial to his argument is the subtle distinction

between doing something intentionally and intending to do something. The former case might be phrased as deliberately doing an action, while intending to do something means one may not be performing the action in order to achieve it. Cohen and Levesque [5], on the other hand, developed a logic in which intention is defined. They define the notion of individual commitment as *persistent goal*, and an intention is defined to be a commitment to act in a certain mental state of *believing throughout* what he is doing.

Thus to provide a definition of attitude that is concrete enough for computational purposes, we model attitude using goals, beliefs, intentions and commitments. From the Fishbein's [6] definition it is clear that when an attitude is adopted, an agent has to exhibit an appropriate behaviour (predisposition means *behave in a particular way*). The exhibited behaviour is based on a number of factors. The most important factor is goal or several goals associated with the object. During problem solving, an agent in order to exhibit behaviour may have to select from one or several goals depending on the nature of the dynamic world.

In a dynamic multiagent world, the behaviour is also based on appropriate commitment of the agent to all unexpected situations in the world including state changes, failures, and other agents' mental and physical behaviours. An agent intending to achieve a goal must first commit itself to the goal by assigning the necessary resources, and then carry out the commitment when the appropriate opportune comes. Carrying out the commitment typically involves planning for the goals, and executing the plans where planning involves decomposing complex goals, and executing the plans ultimately involve executing the primitive physical and communicative actions. Second, if the agent is committed to executing its action, it needs to know how weak or strong the commitment is. If the commitment is weak, the agent may not want to expend too much of its resources in achieving the execution. Thus, the agent needs to know the degree of its commitment towards the action. This degree of commitment quantifies the agent's *attitude* towards the action execution. For example, if the agent considers the action execution to be *higher importance* (an attitude towards the action), then it may choose to execute the action with greater degree of commitment; otherwise, the agent may drop the action even when it had failed at the first time. Thus, in our formulation, an agent when it performs an activity, since the activity is more likely that it will not succeed in a dynamic world, agents will adopt a definite attitude towards every activity while performing that activity. The adopted attitude will *guide* the agent in responding to failure situations. Also the behaviour must be consistent over the period of time during which the agent is holding the attitude. Thus attitudes, once adopted, must persist for a reasonable period of time so that other agents can use it to predict the behaviour of the agent under consideration. An agent cannot thus afford to change its attitude towards a given object too often, because if it does, its behaviour will become somewhat like a reactive agent, and its attitude may not be useful to other agents. Once an agent chose to adopt an attitude, it strives to maintain this attitude, until it reaches a situation where the agent may choose to drop its current attitude towards the object and adopt a new attitude towards the same object. We thus, define attitude as follows: *An agent's attitude towards an object refers its persistent degree of commitment towards achieving one or several goals associated with the object, which give rise to an overall favourable or unfavourable behaviour with regard to that object.*

We thus can view goals, beliefs, intentions and commitments as primitive forms of attitudes. In a dynamic world, beliefs are formed from direct observation, messages

received from other agents, and through inference. From beliefs, arise attitudes. Depending on what the beliefs are the attitudes may take different polarity. Thus, the attitudes sometimes may be favourable and sometimes unfavourable, but what matters is the overall affect towards the object involved in the attitude. Accordingly as intentions are commitments to persistent goal [5], we can model intentions as always a favourable attitude. While in general attitudes can be specific about how soon or late a chosen activity must begin and end, intention implicitly demands that the activity start within a reasonable time. Also, once an adopted intention is dropped, it cannot be readopted; and if it is, it is usually a different intention. Thus, an adopted intention refers to a continuous and uninterrupted mental state. As opposed to this, attitudes may be continuous or intermittent. That is, an attitude by its definition may demand a mental behaviour that is distributed over time. Also from the above definition it is clear that, an attitude towards an object has the character of a commitment towards that object. In this sense it represents a dynamic assessment of a given situation with reference to an end. One fundamental consequence of such an assessment is that certain facts become relevant, others less relevant and that certain data become crucial, others less important; consequently, certain old plans may have to be revised.

### 3.2 Fire World Attitudes

A fire world is dynamic, hostile, typically multiagent and an agent in this world responds to the events both reactively and deliberately. The agents need to act as a team in order to solve problems in this world. The overall behaviour of a team of agents is dependent upon the attitudes of the agents towards the team as well as on the attitudes of the agents towards the group and the society it is part of. In this section, we will present a set of attitudes specific to the fire world domain. Although we only enlist the attitudes that are needed in the fire world domain, most of these attitudes are generic in nature and can be used in other domains as well with some modifications.

The agents in the fire world have attitudes towards three types of objects: (i) physical objects (ii) mental objects (iii) and agents.

**Physical Objects:** As described above, the fire world consists of physical objects like tables, chairs, walls, rooms and buildings. When the fire is set in the fire world, new objects such as fire, ash, smoke, half-burnt objects etc. are created. Agents in the fire world in response to the fire adopt attitudes towards each of these objects. The adopted attitudes guide the behaviour of agents. We consider the following physical objects: {chair, table, wall, room, shelf, tree, chemical, fire, smoke, ash}

**Mental Objects:** Agents in the fire world hold attitude towards mental objects as well. The list of mental objects we have considered in our fire world is as follows: {beliefs, goals, plans, time line structure, actions, rules}.

**Agents:** There are two types of agents in our domain i.e. victim agents and fire-fighting agents. The attitudes discussed in this section may be individual or collective. Individual attitudes are explicit and collective attitudes are implicit.

### 3.2.1 Attitudes towards Physical objects

Fire is perhaps the most important object an agent has to deal within a fire world. Thus, it is necessary to have a set of attitudes towards fire that the agent can adopt at appropriate situations.

#### (1) **Small-Fire** (Fire *f*)

This is an attitude towards fire and it implies that the agent considers the fire to be small. The fire-fighters holding this attitude believe that putting out this fire is an easier task as compared to agents holding attitudes such as **Medium-Fire** and **Large-Fire** (see later part of this section). Even if the fire is considered small, the agents will always have **dangerous-fire** attitude attached to it, because the fire always creates dangerous situations, spreads out quickly and can thus become medium fire or large fire.

*Name of attitude: Small-Fire*

*Description of object:* (1) name of the object: *f*.

(2) model of the object :

(2.1) Location of squares on fire: *sq1, sq2*.

(2.2) Number of squares on fire:  $\leq$  *smax*

*Evaluation:* unfavourable /\* victim agents and fire-fighting agents \*/

*Behaviour:* This attitude produces the put out fire behaviour by the fire-fighting agent.

In order to put out the fire, agents go to the fire and hose water on the squares, which are on fire. Then the fire-fighters will check if the fire is put out or not, if not then they will spray water again. Thus behaviours produced in case of small fire are:

*goto(sq), spray-water(sq), checkfire-spray(sq) etc.*

*Persistency:* if small fire becomes a medium fire, but the actions in the immediate future include putting out at least a few squares, then the fire will still be considered as small; otherwise change to other attitudes is necessary,

*Concurrent attitudes: Dangerous-Fire*

*Type:* can be *individual* as well as *collective*.

#### (2) **Medium-Fire** (Fire *f*)

This attitude is adopted when the agent considers the fire to be of medium size. The attitude can be generated directly as a medium fire breaks out in a region or it changes from small fire to become medium fire.

*Name of attitude: Medium-Fire*

*Description of object:* (1) Name of the object: *f*.

(2) Model of the object:

(2.1) Location of squares on fire: *sq1, sq2, ..., sqn*.

(2.2) Number of squares on fire:  $\leq$  *mmax*

*Evaluation:* unfavourable /\* victim agents and fire-fighting agents \*/

*Behaviour:* This attitude produces the put out fire behaviour by the fire-fighting agent.

The put out fire in case of the medium fire is same as small fire, though it may take more time as compared to the large fire.

*/\* Fire-fighting behaviour as specified by the rule set above \*/*

*Persistency:* hold on to the attitude as long as the fire is of medium size.

*Concurrent attitudes: Dangerous-Fire*

*Type:* Can be *individual* and *collective*.

(3) **Large-Fire** (Fire *f*)

This attitude is adopted when the agent considers the fire to be of larger size. While holding this attitude, an agent abandons the team activity and the team. However, when the large fire transforms into medium or small fire, the team may be formed again.

*Name of attitude:* **Large-Fire**

*Description of object:* (1) Name of the object: *f*.

(2) Model of the object:

(2.1) Location of squares on fire:  $sq_1, sq_2, \dots, sq_n$ .

(2.2) Number of squares on fire:  $> m_{max}$ .

*Evaluation:* unfavourable.

*Behaviour:* The fire-fighters quit whatever they are doing and run towards the exit. The fire-fighting team also dissolves whenever there is large fire.

*Persistence:* hold on to this attitude unless the fire has been extinguished.

*Concurrent attitudes:* **Dangerous-Fire, Escape, Team-uniform**

*Type:* can be both *individual* and *collective*.

(4) **Dangerous-Fire** (Fire *f*)

The fire is harmful to both for the agents as well as for the physical world. So whenever, there is a fire the victims as well as the fire-fighters will hold this attitude. In the restricted world we have considered in this thesis, a fire is considered dangerous if there are chemicals or LPG's involved in it. Thus a fire near a chemical will be considered dangerous. But if the chemicals have been moved away, the fire is no longer considered dangerous.

*Name of attitude:* **Dangerous-Fire**

*Description of object:* (1) Name of the object: *f*

(2) Model of the object:

Location of squares on fire:  $sq_1, sq_2, \dots, sq_n$ .

*Evaluation:* unfavourable

*Behaviour:* Perform behaviour that will transform the fire from dangerous to normal i.e. remove all chemicals away. If the fire occurs in LPG room, abandon all activities and escape.

*Persistence:* hold on to this attitude until the agents have made their exit to the safe place. Once the chemicals are removed, this attitude will be dropped and a different attitude will be adopted.

*Concurrent attitudes:* coexists with all the other fire attitudes.

*Type:* can be *individual* as well as *collective*.

(5) **Warning** (Chemical *c*)

This attitude is invoked, whenever the firefighting team has to deal with the chemicals. The chemicals in the fire world are more combustible as compared to other physical objects in the fire world. So whenever, the fire-fighters are in the fire world and are dealing with chemicals, they try to warn each other or victims about the presence of chemicals. Therefore, the other concurrent attitudes invoked with this attitude are **Dangerous-fire** and **Cautious**.

### 3.2.2 Attitudes towards Mental Objects

The second type of attitudes is attitudes towards mental objects. In our fire world, we let agents have attitudes towards plans, rule base, goals and actions.

#### (1) *Cautious* (Plan $p$ )

By adopting this attitude means that when the agents are executing team plan, it will do every activity very “carefully”. (See basic agent behaviour below).

*Name of Attitude: Cautious*

*Description of Object:* (1) Name of the object:  $p$ .

(2) Model of the object: /\* a model of the plan \*/

A copy of the plan  $p$  loaded in TLS (section 4.1).

*Basic agent behaviour:* When the agents are doing the team activity, they will spend additional effort to find out changes in the world states and other agents’ activities. They will also check whether other agents are still in the team or the team plans are still valid and so on. Agents doing team activity cautiously often takes more time and consume more resources than in the normal mode.

*Persistence:* hold on this attitude as long as the agents execute the plan cautiously.

*Evaluation:* favourable

*Concurrent attitudes:* all the fire related attitudes.

*Type of Attitude:* can be *individual* as well as *collective*.

#### (2) *Casual*(Plan $p$ )

By adopting this attitude, the agents will just execute the team plan without “caring” about the consequences.

*Name of Attitude: Casual*

*Description of Object:* (1) Name of the object:  $p$ .

(2) Model of the object: /\* a model of the plan \*/

A copy of the plan  $p$  loaded in TLS (section 4.1).

*Basic agent behaviour:* With this attitude, the agents just keep on doing whatever they are doing and don’t worry about the consequences of team plan or action.

*Persistence:* hold on this attitude as long as the agents keep on executing the plan casually.

*Evaluation:* unfavourable /\* victim agents and fire-fighting agents \*/

*Concurrent Attitudes:* all the fire related attitudes.

*Type of Attitude:* can be *individual* as well as *collective*.

#### (3) *Slowly*(Plan $p$ )

By adopting this attitude, the agents will take more time to complete the execution of the plan. For example, when going from one place to another in a fire world the agents will move very “slowly”.

*Name of Attitude: Slowly*

*Description of Object:* (1) Name of the object:  $p$ .

(2) Model of the object: /\* a model of the plan \*/

A copy of the plan  $p$  loaded in TLS (section 4.1).

*Basic agent behaviour:* With this attitude, the agents take more time to do the team tasks.

*Persistence:* hold on this attitude as long as the agents keep on executing the plan slowly.

*Evaluation:* unfavourable

*Concurrent attitudes:* all the fire related attitudes.

*Type of Attitude:* can be *individual* as well as *collective*.

#### (4) **Quickly**(Plan $p$ )

With this attitude, the agents try to complete the execution of the plan as quickly as possible eg. The fire-fighters in the fire world try to go to the destination or place where fire has set as quickly as they can.

*Name of Attitude:* **Quickly**

*Description of Object:* (1) Name of the object:  $p$ .

(2) Model of the object: /\* a model of the plan \*/

A copy of the plan  $p$  loaded in TLS (section 4.1).

*Basic agent behaviour:* With this attitude, the agents take very less time to do the team task

*Persistence:* hold on this attitude as long as the agents keep on executing the plan quickly.

*Evaluation:* favourable

*Concurrent attitudes:* all the fire related attitudes.

*Type of Attitude:* can be *individual* as well as *collective*.

#### (5) **Safe** (Situation $s$ )

An attitude the agent holds when it considers the situation  $s$  to be safe.

*Name of Attitude:* **Safe**

*Description of Object:* (1) Name of the object:  $s$ .

(2) Model of the object:

A model of all the physical and mental objects of the world.

*Basic agent behaviour:* invokes team behaviour rules e.g. going to a destination.

*Persistence:* hold on this attitude as long as the situation in the fire world is safe.

*Evaluation:* favourable

*Concurrent attitudes:* all the fire related attitudes.

*Type of Attitude:* can be *individual* as well as *collective*.

#### (6) **Unsafe** (Situation $s$ )

If the team agent has **unsafe** attitude, the agent will abandon the team activity as specified below.

*Name of Attitude:* **Unsafe**

*Model of Object:* (1) Name of the object:  $s$ .

(2) Model of the object:

A model of all the physical and mental objects of the world.

*Basic agent behaviour:* The agent holding this attitude abandons the team activity.

*Persistence:* hold on this attitude as long as the agents keep on executing the plan casually.

*Evaluation:* unfavourable

*Concurrent attitudes:* all the fire related attitudes.

*Type of Attitude:* collective

(7) **Evacuate** (Plan  $p$ )

This attitude is generated when the fire has gone out of control in the room and the fire-fighters ask the victim agents to go out of the room. The plan  $p$  is a plan for evacuation and it is executed with an attitude that it is an evacuation plan.

*Name of attitude:* **Evacuate**

*Description of object:* (1) Name of the object:  $p$ .

(2) Model of the object:

(2.1) Time of execution: 1,2 , ...n

(2.2) Number of actions: n

*Behaviour:* The plan executional behaviour of the agent for this plan must be such that the execution is successful even when the world is dynamically changing.

The plan executional behaviour is as follows:

- abandon all team related attitudes and activities.
- If an action failure occurs, replan for the action, and execute it.
- If replanning fails, ask for assistance.

*Persistence:* hold on to the attitude until the evacuation is achieved.

*Concurrent attitudes:* exists with attitudes like **Dangerous-fire** and **Large-Fire**.

*Type:* individual.

### 3.2.3 Attitudes towards other agents.

The third type of attitude is attitudes towards other agents, which includes victim agents as well as fire-fighting agents.

(1) **Rescue** (Agent  $A_i$ )

This attitude helps the team to save the trapped agent in the fire world.

*Name of Attitude:* **Rescue**

*Description of Object:* (1) Name of the object:  $A_i$ . / \* object is an agent \*/

(2) Model of the object:

(2.1) Location of Agent:  $s_{qi}$

(2.2) Mental state of agent: set of goals the victim wants to achieve  $\{G_1, .. ,G_n\}$ .

*Basic agent behavior:* The team asks the agent to move to a particular safe place and remove the obstacles around it.

*Persistence:* holds on this attitude as long as the victim agent is being rescued.

*Evaluation:* favourable

*Concurrent attitudes:* all the fire attitudes.

*Type of Attitude:* collective

(2) **Ignore** (Agent  $A_i$ )

With this attitude, the team agents don't care about the victim agents in the fire world and only worry about the welfare of the team or themselves.

*Name of Attitude:* **Ignore**

*Description of Object:* 1) Name of the object :  $A_i$ . / \* object is an agent \*/

(2) Model of the object:

(2.1) Location of Agent: sqi

(2.2) Mental state of agent: set of goals the victim wants to achieve  $\{G_1, \dots, G_n\}$ .

*Basic agent behavior:* The fire-fighting team ignores the victim agent and keeps on doing its task.

*Persistence:* holds on this attitude as long as the victim agent is being ignored.

*Evaluation:* unfavourable

*Concurrent attitudes:* all the fire attitudes.

*Type of Attitude:* can be individual as well as collective

### 3.4 Attitudes in a Team

The attitudes of an agent existing in a team consist of attitude towards team as well as attitude towards team activity. At any time, an agent may be engaged in one of the basic team activities i.e. team formation, team maintenance, and team dissolution. Instead of modelling these basic activities as tasks to be achieved, we have chosen to model them as attitudes. When the agents are forming a team, the agents forming a team have attitude called **Team-form**. This attitude guides the agent to form a team. As discussed above when the agents are in the team they should have the **Team** attitude to do a team activity. Similarly, when the agents are dissolving the team, they should have **Team-unform** attitude. The **Team** attitude can be represented in the form of individual attitudes towards the various attributes of the team i.e. team definition, team methods, team rule base, team mutual belief and responsibility. We can have a number of attitudes towards the each of these attributes, which will guide the agents to solve a team problem. For example, attitude towards team definition can be **Periodic-team-maintenance** or **Situation-team-maintenance**. The attitudes towards the responsibility attribute can be *social* attitudes or *group* attitudes. The attitudes towards the social responsibility are *social* attitudes, which are individual attitudes directed toward social objects. The social attitudes express a social relationship between agents. Through the social attitudes practically all the social conventions are consummated. The examples of social attitudes in our team model are **Wait** or **Help**. The *group* attitudes help the agents in coordinating the various group activities. Thus the agents in the team have to coordinate their various activities as a part of their group responsibility. The group activity in the team can be fully-coordinated, partially coordinated and non-coordinated. So the example of group attitude in our model is **Full-coordination**. The following are the computational models of the few *individual* as well as *collective* attitudes present in our team model.

(1) **Team** (A1, A2)

This attitude is invoked when the agents are in a team state. This attitude guides the agents to perform the appropriate team behaviours.

*Name of Attitude:* **Team** /\* implicit attitude \*/

*Description of Object:* (1) Name of Object:  $A^T$  (section 5.2)

(2) Model of Object:  $\{A_1, A_2 \mid A_i \text{ is an agent}\}$

*Basic agent behaviour:* team behaviour specified by  $R^T$

*Evaluation:* favourable

*Persistence:* This attitude persists as long as the agents are able to maintain it.

*Concurrent attitudes:* all attitudes towards physical and mental objects in the domain.

*Type of Attitude:* collective.

**(2) Team-form** ( $A_1, A_2$ )

This attitude is invoked when the agents have to form a team to solve a complex problem.

*Name of Attitude:* **Team-form**

*Description of Object:* (1) Name of Object: set of agents.

(2) Model of Object:  $\{A_1, A_2 \mid A_i \text{ is an agent}\}$

*Basic agent behaviour:* invokes team formation rules (section 5.2.1.1).

*Evaluation:* favourable

*Persistence:* The agent holds this attitude as long as it believes that a team formation is possible.

*Concurrent attitudes:* All attitudes towards physical and mental objects in the domain.

*Type of Attitude:* individual

**(3) Team-uniform** ( $A_1, A_2$ )

This attitude helps the agents to uniform the team due to a very risky situation.

*Name of Attitude:* **Team-uniform**

*Model of Object:* (1) Name of Object: set of agents.

(2) Model of Object:  $\{A_1, A_2 \mid A_i \text{ is an agent}\}$

*Basic agent behaviour:* invokes team-uniformation rules.

*Evaluation:* The evaluation of this attitude is unfavourable, because it results in uniformation of the team.

*Persistence:* The agent holds this attitude as long as it believes that the team structure cannot be supported in the current situation.

*Concurrent attitudes:* All attitudes towards physical and mental objects in the domain.

*Type of Attitude:* individual

**(4) Periodic-team-maintenance** ( $A_1, A_2$ )

This individual attitude towards set of agents invokes the team definition ( $DEF^T$ ) goal periodically. To invoke the team definition periodically, the agent has to execute team definition goal rules periodically.

*Name of Attitude:* **Periodic-team-maintenance**

*Description of Object:* (1) Name of Object: set of agents.

(2) Model of Object:  $\{A_1, A_2 \mid A_i \text{ is an agent}\}$

*Basic agent behaviour:* The team agent invokes team definition ( $DEF^T$ ) rules periodically.

*Evaluation:* favourable /\* towards team \*/

*Persistence:* This attitude is available only when the team definition goal is invoked.

*Concurrent attitudes:* attitudes towards other physical and mental objects in the domain.

*Type of Attitude:* individual

**(5) Situation-team-maintenance** ( $A_1, A_2$ )

If the attitude towards the team definition ( $DEF^T$ ) is situation dependent, the agent would invoke the team definition goal whenever there are major changes in the world.

*Name of Attitude: **Situation-team-maintenance***

*Description of Object:* (1) Name of Object: set of agents.

(2) Model of Object: {A1, A2 | Ai is an agent}

*Basic agent behaviour:* invokes team definition rules whenever there is a change in the state of the world.

*Evaluation:* favourable

*Persistence:* This attitude exists only when the team definition rules are fired.

*Concurrent attitudes:* attitudes towards other physical and mental objects in the domain.

*Type of Attitude:* individual

(6) **Wait** (A1, A2)

Whenever the agents have this attitude, the team agents try to wait for other agents while sharing the common resources in order to fulfil the social obligations or commitments.

*Name of Attitude: **Wait***

*Description of Object:* (1) Name of Object: set of agents

(2) Model of Object: {A1, A2 | Ai is an agent}

*Basic agent behaviour:* wait for the other agent

*Evaluation:* favourable

*Persistence:* This attitude persists as long as the agents are fulfilling the social obligations.

*Concurrent attitudes:* attitudes towards other physical and mental objects in the domain.

*Type of Attitude:* collective

(7) **Help** (Ai)

This attitude is invoked when agent  $A_j$  believes  $A_i$  requires help and  $A_j$  can afford it.

*Name of Attitude: **Help***

*Description of Object:* (1) Name of Agent: Ai

(2) Agent State:

(i) physical state of Ai

(ii) model of Ai's TLS (section

*Basic agent behaviour:* invokes a set of rules specifying the helpful behaviour of the agent.

*Evaluation:* favourable /\* favouring agent  $A_i$  \*/

*Persistence:* This attitude persists as long as the agents are fulfilling the social obligations.

*Concurrent attitudes:* attitudes towards social and group responsibilities eg. **Wait**.

*Type of Attitude:* collective

(8) **Full-coordination** (Ai)

This attitude is adopted as a part of agent's group commitments. With this attitude, the agents try to fully coordinate the group activity i.e the start and end time of doing the activity of two agents is the same.

*Name of Attitude: Full-coordination* /\* implicit attitude \*/

*Description of Object:* (1) Name of Agent: Ai

(2) Agent State:

(i) physical state of Ai

(ii) model of Ai's TLS (section

*Basic agent behaviour:* helps the agent to fully co-ordinate their activities.

*Evaluation:* favourable

*Persistence:* This attitude persists as long as the agents exist as a group.

*Concurrent attitudes:* attitudes towards social and group responsibilities eg. **Wait, Help.**

*Type of Attitude:* collective

In this section, we have proposed a number of attitudes that play essential roles in the fire domain. There can be a lot of more attitudes in the fire world than the few attitudes we described above, but due to the computational and implementation difficulties we have considered the above set of attitudes only. We hope that the list of attitudes we have proposed above are adequate for specifying the complex problem solving behaviours of the agents in the fire world.

#### 4. Problem Solving by Team Agent

The problem solving by the team agent ( $A_T$ ) in our fire world involves the firing of rules in the team rule base. These rules are constrained by the conditions of the world under which they are invoked. The basic behaviours depicted in a fire world by the team agents are putting out fire, rescuing victims, saving property and team dissolution etc. These team rules result into generation of various team goals, which are decomposed into sub goals. The various goals and sub-goals ultimately result in team behaviours. The following are a few team rules invoked in a fire world:

Rule 1: (rule name: **behaviour for small fire**

Condition: fire is small

--> Behaviour: 1. *go-to(square)*,  
2. *spray-water(square)*,  
3. *check-fire-spray(square) ...*)

Rule 2: (rule name: **behaviour for medium fire**

Condition: fire is medium fire

--> Behaviour: 1. *go-to(square)*,  
2. *spray-water(square)*,  
3. *check-fire-spray(square)..*)

Rule 3: (rule name: **behaviour for large fire**

Condition: fire is large

--> Behaviour: 1. *team-dissolution(team)*,  
2. *move-towards-exit(agent)...* )

Rule 4: (rule name: **behaviour of fire-fighters for victim agents**

Condition: Agents trapped in fire in a room or building.

--> Behaviour: 1. *go-to-victim(victim)*,  
2. *put-out-fire(victim)*,  
3. *spray-water(square)*,

4. *check-fire-spray*(fire),
5. *escape*(victim)... )

Rule 5: (rule name: **behaviour for moving objects**

Condition: Objects are to be saved from catching the fire.

- > Behaviour: 1. *go-to-object*(object),  
2. *pick-object*(object)  
3. *move-with-object*(object)  
4. *drop-object*(object) ...)

#### 4.1 Agent Architecture

Having discussed how attitudes play an important role in team problem solving, we now propose agent architecture and discuss its features, which guide the agents in team problem solving. The agent architecture (Figure 3) consists of the following modules:

*Sensor*: The function of sensor is to sense the changes in the world and accordingly update its world model.

*World Model and Model of Other Agents*: The world model contains a hierarchical description of the world. For example, agents in the fire world view world at different levels such as campus, buildings and rooms etc. The model of other agents consists of a partial knowledge of the other agents, which is very useful in teamwork.

*Attitude Generator*: An agent responds to events occurring in the world by generating attitudes through the attitude generator. When the sensor senses the changes in the world, the agent updates its world model and forms the various beliefs. From the beliefs the attitudes towards the team as well as towards the activity are formed. The attitude generator is a module, which generates attitudes to specify the right type of behaviour required for team activity as well as for the maintenance of team.

*Goal Generator and Goal Pool*: The function of goal generator is to generate different goals under different conditions. The goals generated are stored in goal pool for further execution. In the fire world many types of goals are generated eg. main goals which are relevant for achieving team task, specific goals which assist agents to get out of a dead end, goals to escape a dangerous situation, and social convention type of goals such as, helping the other agents to pick up objects.

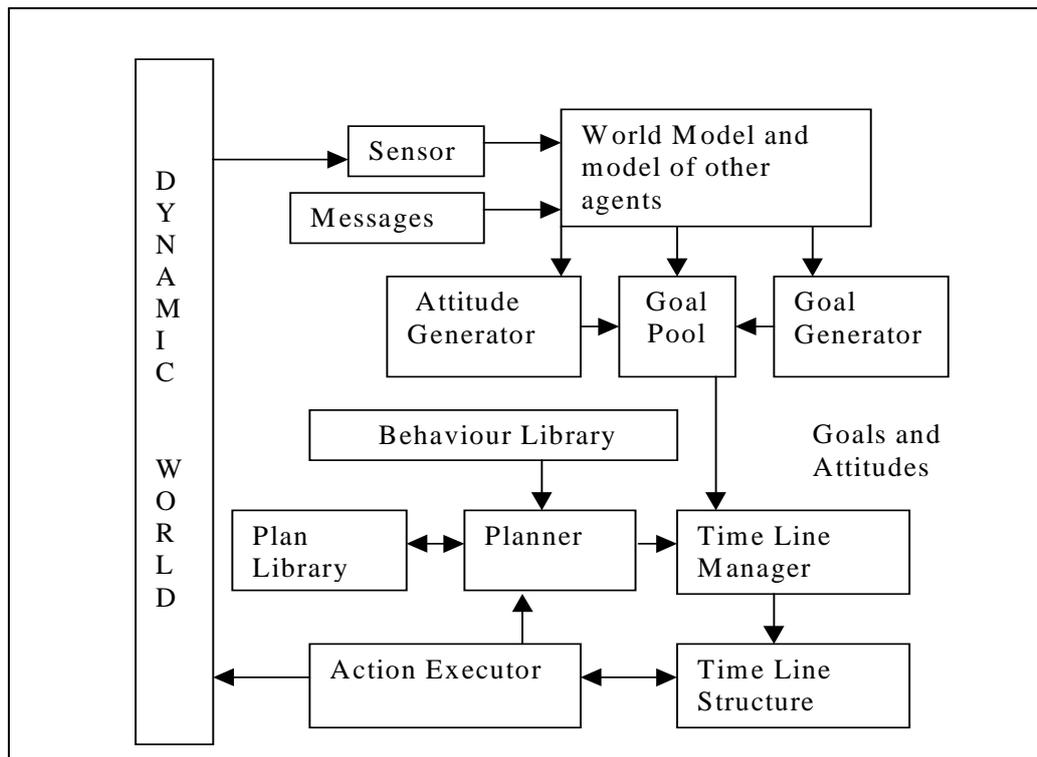


Figure 3: Agent Architecture

*Behaviour Library:* Each attitude has a set of associated behaviours. In our implementation, these behaviours are specified by a set of rules and are stored in the behaviour library.

*Planner:* The function of planner is to derive a plan for a given goal from plan library. For example, when a new fire is reported, the planner retrieves an appropriate plan from the plan library and places it on the time line. Information about the resources and sensory input are stored in the world model and is used to help the agent to select the appropriate plan. At any time during this process, sensory data can trigger reactive actions, which will cause the team planner to modify its plan (such as moving away from the fire ahead) by incorporation of some rules.

*Time line Structure:* The time line structures (TLS) are the nuts and bolts for defining and implementing the issues that are identified by the team. A time line structure is a way or method that the agent chooses and agrees on to do its task and achieve their goal. The purpose of a time line structure is to schedule the team activities. All the team activities are scheduled in a particular sequence or a pattern. A team agent picks up the most recent goal/activity, expands it dynamically into low level details and inserts them into TLS for execution. When agents form a team, problems emerge regarding the representation and execution of actions. The team time line contains the details of the overall task to be performed by the team. The overall task is further decomposed into subtasks, which again might be further decomposed into individual agent subtasks. Thus the team time line module helps to formalise complex team plans into individual team actions. The team time-line contains the time-line structure (TLS) for the team as well as the individual agents

The team plan contains both an abstract structure and a set of complex mental attitudes. But the time line structures (TLS) are representations of the activities of the team it performs such as team plan, communication or resource allocation. Each time line entry instantiates a goal or action, binding its variables and adding the temporal constraints that relate it to other actions the agent has chosen to execute. These temporal constraints are determined by the attitudes of the agents towards that action. The actions on the time line (TLS) can be physical, communicative and mental actions. In general each time line structures consists of primitive actions, abstract actions, team goals, team rules, conditional actions and loops.

*Time Line Manager:* The time line manager then places the goal generated or the team plan on the time line structure (TLS).

*Action Executor:* The function of executor is to execute the time line structure. The plan loaded on the time line can be modified, suspended or cancelled according to the attitude of team agent towards the team activity.

#### 4.2 Execution of Team TLS

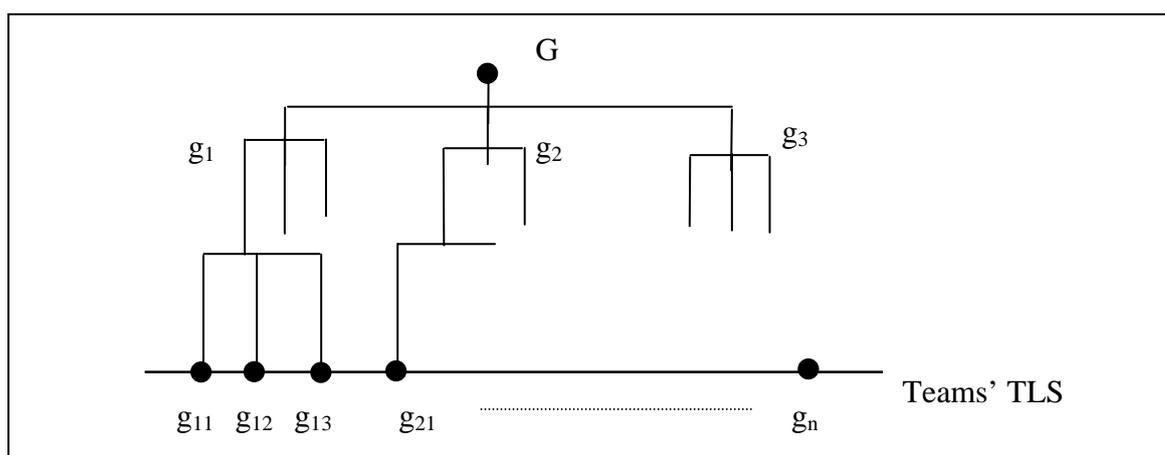


Figure 4: Expansion of Team Goals on Team TLS

The team time line structure (TLS) specifies a series of team activities that the agent needs to perform in order to achieve or maintain the team goal. A team goal  $G_T$  to be achieved must be decomposed into goals  $g_1, g_2, g_3, \dots, g_n$  (sub-goals) where  $G_T$  is the top level goal, while  $g_1, g_2, g_3 \dots g_n$  are sub-goals of achieving  $G_T$  in particular circumstances. The decomposition operator used typically results in decomposing a given goal into temporally ordered “smaller” goals. Therefore, the expanded time line structure (TLS) will look like as shown in figure 4. The time line structure will keep on extending like this until all the goals can't be further divided into sub-goals ( $g_{11}, g_{12}, g_{13}, \dots$ ) and ultimately results into primitive actions. For example, when a fire is reported, a goal called "Put-Out-Fire" is generated by the goal generator. The goal generator then retrieves a corresponding plan from the plan library and passes it to the time line manager. The time line manager loads the appropriate plan for the team activity on the time line and the plan execution is then initiated.

### 4.3 Basic Team Activities in a Fire World

In the fire world, the main task of the team of fire-fighting agents is to put out the fire. While putting out the fire the fire-fighting agents have to do lot of other tasks like moving to a destination, saving the property by moving the objects of the fire world to a safe place and finally rescuing the victim agents. As explained in Chapter 4, to do these team tasks every agent should be provided with a method to do it. It means that every fire-fighting agent should have knowledge and capability of doing that particular task. For example, we know that there are a number of methods for putting out the fire. But in our fire world, the fire-fighting agents extinguish the fire by hosing the water. So each fire fighter in the team should know how to pour water on a burning object. The fire-fighting agents should also know how to move to a destination together, save the property and rescue the victims. Thus in our domain, team goal generator generates four main goals for the team agents in case of fire. These goals are moving to a destination, to put out fire, move chemicals and rescue victims. These goals produce appropriate team behaviours, which are however constrained by different team and individual attitudes in a fire world.

#### 4.3.1 Move to a Destination /\* Plan Based Behaviour \*/

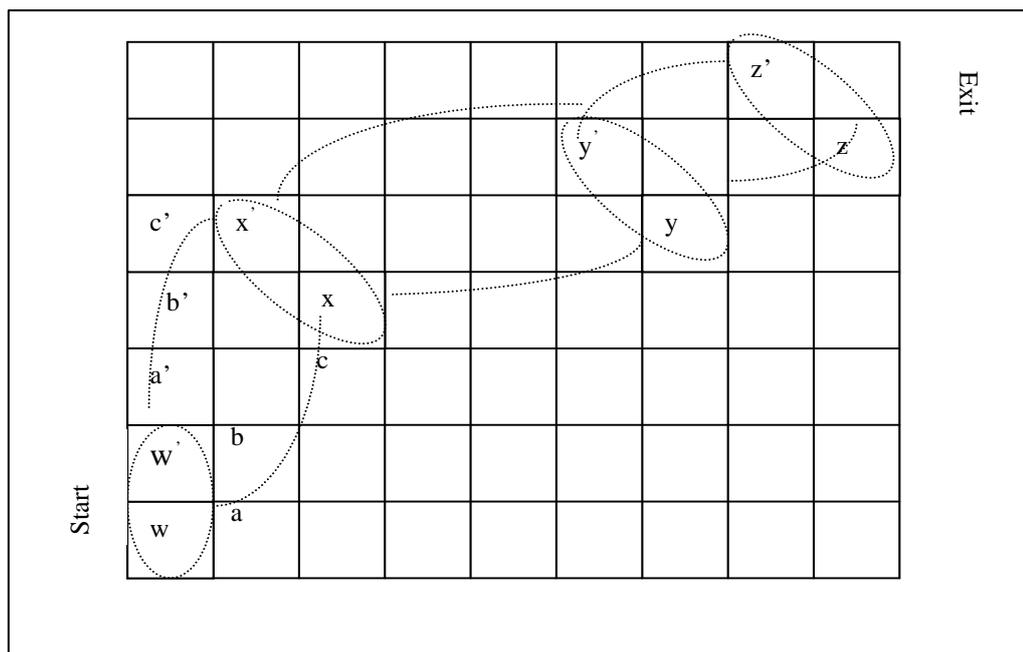


Figure 5: Agents in a Team Moving to a Destination Without Carrying Hose

In our domain, moving from one place to another is a major activity. In almost every task, the team of agents have to move from one place to another. The fire fighters in order to put out fire, have to go from one location to another. The fire-fighters in a fire world move to a destination either carrying a hose or without a hose. Moving to a destination carrying a hose is a fully coordinated activity, while moving to a destination without a hose requires partial co-ordination.

To move to a destination without hose as a team, we assume that team agents meet at  $n$  intermediate points. This means that at every node, each agent in team starts by sending

the START message to other agent. Then each agent does the individual navigation to reach the intermediate point. When it reaches the intermediate point it sends END message to the other agent. In this way the agents in a team reach destination while meeting at intermediate points.

Consider a team consisting of agents  $A_1$  and  $A_2$ . The team goal ( $G_T$ ) is to move towards the exit i.e. ( $G_T = \text{move-to-exit (team)}$ ). To move to a destination as a team we say that team agents meet at  $n$  intermediate points. This means that at every node team starts by sending the START message to other agent. Then each agent does the individual navigation to reach the intermediate point. When it reaches the intermediate point it sends END message to the other agent. Thus the overall team plan looks like as follows:

$\text{TeamMove(pos)} = [\text{TeamStart(Agent), Ind-Navig(Agent), TeamEnds(Agent)}]$

In our case (Figure 5), the team meets at three intermediate points. So the team activity  $G_T$  is divided into three sub activities i.e.  $g_{1T}$ ,  $g_{2T}$ ,  $g_{3T}$  or  $G_T = [g_{1T}, g_{2T}, g_{3T}]$ . Where  $g_{1T} = \text{go-to-sq}(x, x')$ ,  $g_{2T} = \text{go-to-sq}(y, y')$ , and  $g_{3T} = \text{go-to-sq}(z, z')$ . The agent  $A_1$  is at the location  $w'$ , so the individual navigation while going from  $w'$  to  $x'$  is:

$\alpha_{1A_1} = [\text{go-to-sq}(a'), \text{go-to-sq}(b'), \text{go-to-sq}(c'), \text{go-to-sq}(x')]$

The agent  $A_2$  is at the location  $w$ , the individual navigation of the agent while going from  $w$  to  $x$  is:

$\alpha_{1A_2} = [\text{go-to-sq}(a), \text{go-to-sq}(b), \text{go-to-sq}(c), \text{go-to-sq}(x)]$

Similarly, the agents will do the individual navigation while going from  $x'$  to  $y'$ ,  $y'$  to  $z'$ ,  $x$  to  $y$ , and  $y$  to  $z$ .

Moving to a destination without hose is totally different from moving to a destination with hose. The moving to a destination with hose is coordinated individual activity, which involves first going to the hose then carry the hose to destination where it is required. The overall plan for moving to a destination with hose is as follows:

$\text{TeamMove(hose)} = [\text{GotoHose(sq), CarryHose(hose)}]$

Going to the square where hose is kept is same as the team moving to a destination without hose. Carrying hose is coordinated by the two firefighters  $FF_1$  and  $FF_2$  in a way that  $FF_2$  follows  $FF_1$ . Suppose  $FF_1$  and  $FF_2$  going to the  $\text{pos}(sq1, sq2)$ . The firefighter  $FF_1$  will first perform the mental activity of planning to go to  $sq1$  i.e.  $\text{plan}(sq1)$ , then it will execute the plan i.e.  $\text{exec(plan)}$ . Since  $FF_2$  is following  $FF_1$ , the sequence of actions in case of  $FF_2$  is as follows: Wait, RetrievePlan( $sq1$ ), Exec(plan), Plan( $sq2$ ), Exec(plan). The problem encountered with this solution is that though it is individual activity at lower level, still both firefighters have to synchronize with each other to carry hose.

#### 4.3.2 Put Out Fire /\*Rule Based Behaviour \*/

The method we used to put out fire assumes that both the agents in the team have same attitude towards the severity of fire. Inter-agent conflicts due to varied assessment and decision making are beyond the scope of thesis. However, in our problem solving methods we deal with conflicts due to shared resources and obstacle with social and group attitudes. For example, when one agent is blocking the door or passing through

door, the other agent will wait till the other agent has passed through door as a part of its social obligations.

We suggest a rule based solution for the team to put out the fire. The different rules are fired under different situations. We consider different types of fires i.e. small, medium and large. The set of rules for the different types of fires are given in section 4. These rules explain how the fire-fighting agents will behave under fire conditions. To put the fire a team first chooses the position from which it will put out the fire. Then it will move to that position. Then it will choose at which squares the fire should be put out. Then it will hose water on that squares to put out the fire.

Rule : (rule name: **behaviour for putting out fire**

Condition: fire is small or medium

--> Behaviour: 1. *go-to(square)*,  
 2. *spray-water(square)*,  
 3. *check-fire-spray(square)*,  
 4. *if -fire-spray-again(square)*,  
 5. *if-no-fire- go-to-next(square)...*

The above behaviours once invoked are executed in the form of a plan. This plan must be loaded on the TLS and executed. However, each action of the plan actually gives rise to an abstract goal. The behaviour *go-to(square)* involves a mental action of choosing the location to move i.e. *choose(sq)* and then team moving to the location i.e. *TeamMove(agent)*. The behaviour *spray-water(square)* is a primitive action, simply involves the hosing water on the square on fire. The behaviour *check-fire-spray(square)* is a conditional action, checking whether the square on which water is sprayed is on fire or not. If the square is still on fire, the fire-fighting agents will spray again. If the fire on that square is put off, then the fire-fighting agents will go to the next square.

### 4.3.3 Save Property /\*Plan Based Solution \*/

The fire environment is dynamic because everything changes: the size of fire, intensity of fire, the availability and position of fire-fighting objects. The environment is real time in the sense that the fire sets the pace to which the agent must adapt. These characteristics require an agent to have concepts of optimality and efficiency. The agent must reason about the potential effects of its actions and, particularly, about how much time these actions must require.

The unpredictability of the environment requires agents to be flexible, particularly in the way they handle resources. Because the environment is ongoing, decisions about resources have long term effects that constrain actions, require agents their resources intelligently **with or without global perspective**. For example, while moving chemicals out of a room, the intelligent solution to the problem is that put chemicals in a box and move the box out of the room. However, in the fire environment a box may not be available to the fire-fighters (box may be burnt out or too small to carry chemicals). In that case, the fire fighters have to rely on a non intelligent or inefficient solution of picking chemicals one by one and moving them out of room. Thus in a hostile, dynamic world the efficiency is largely dependent on the availability of resources.

In order to save the property, the fire fighting agents have to move the chemicals out of the room. To move the chemicals out of the room, the team will first choose the chemicals to be first taken out of the room. It will then pick those chemicals and then move out of the room. So the plan for moving the chemicals out of room or saving the property is as follows:

TeamMoveChemicals(pos) = [Choose(chemicals), Pick(Chemicals), TeamMove(pos)]

The activity Choose(chemicals) is an abstract action, which is further divided into two sub-actions. The firefighter FF1 decides which chemicals are to be picked by himself as well as by the other firefighter.

Choose(chemicals) = [Choose(FF1), Agree(FF2)]

The activity Pick(Chemicals) is a primitive action which simply involves picking up the chemical by each agent. After picking up the chemicals, the agents will start moving towards the exit of the room. This involves the activity TeamMove(pos), which is basically the team moving to a destination.

#### 4.3.4 Rescue Victims /\* Rule Based Solution \*/

We consider rescuing one victim at a time. To rescue victim, the team agent ( $A_T$ ) will first approach the victim, then put out fire around it and ask it to escape. While moving out, if the fire around the victims appear again, then the fire-fighters will again have to put out the fire around the victims. This process will continue until the victim goes out of the room. The rules for rescuing victim is as follows:

Rule : (rule name: **behaviour for rescuing victims**

Condition: Agents trapped in fire in a room or building.

---> Behaviour: 1. *go-to-victim(victim)*,  
 2. *put-out-fire(victim)*,  
 3. *check-fire(victim)*,  
 4. *if-no-fire- escape(victim)*,  
 5. *if-fire-put-out-fire(victim)... )*

The behaviour *go-to-victim (victim)* is same as the moving to the destination where the victims are located. The behaviour *put-out-fire (victim)* involves the putting out the fire around the victims. The method of putting out the fire around the victim is same as explained above. After putting out the fire around the victim, the firefighting agent will check the fire around victim agent. If the fire is put out, the firefighters will ask the victim agent to escape to the exit. If not, the fighters will try to put out the fire again.

### 5. Problem Solving by Individual Agent ( $A_i$ )

Problem solving by an individual agent in a team involves solving the team goals as well as individual, social and group goals, so that flexible, efficient and coherent team work can be ensured. Thus individual, social and group responsibilities also play an important role in team activity besides the team responsibility. These responsibilities

results in triggering of individual, group and social behaviours along with the team activity related behaviours. These behaviours help the agents in a team to resolve the conflicts at the local level, so that the overall team goal is not affected by the dynamic changes of the world. The following rules depict the individual, social and group behaviours generated under different conditions.

Rule 1: (rule name: **social behaviour of individual agent ( $A_i$ )**)

Condition: The door is blocked by the other agent.

---> Behaviour: 1. *wait(agent)*,  
2. *pass-through-door(agent)...*

Rule 2: (rule name: **group behaviour of individual agent ( $A_i$ )**)

Condition: The door is blocked by objects like table or chair.

---> Behaviour: 1. *go-to-blocked-object(agent)*,  
2. *coordinate(agent)*,  
3. *move-blocked-object(agent)...*

Rule 3: (rule name: **individual behaviour of individual agent ( $A_i$ )**)

Condition: fire is out of control

---> Behaviour: 1. *shout-for-help(agent)*,  
2. *move-towards-exit(agent). .*

Thus the time line of individual agent contains the team goals ( $G_{team}$ ), group goals ( $G_{group}$ ), social goals ( $G_{social}$ ) and individual goals ( $G_{individual}$ ) as shown in figure 6. The team goals are further divided into team formation goals ( $G_{team-formation}$ ), problem solving goals ( $G_{problem-solving}$ ), team maintenance goals ( $G_{team-maintenance}$ ) and the team unformation goals ( $G_{team-unformation}$ ). Similarly group goals ( $G_{group}$ ) are further divided into various group goals ( $G_{group1}$ ,  $G_{group2}$ , ...,  $G_{groupn}$ ) The social goals are divided into various social convention goals etc. The individual goals are divided into different types of individual goals. Besides these goals there are also reactive goals ( $G_{reactive}$ ), communication goals ( $G_{communication}$ ) and the attitude goals ( $G_{attitude}$ ) in a individual agent's TLS.

Rule 4: (rule name: **reactive behaviour for individual agent ( $A_i$ )**)

Condition: The temperature of square is greater than 40 C

---> Behaviour: 1. *jump-off-to-adjacent-square(agent)*,  
2. *jump-again(agent) ...*

Rule 5: (rule name: **communicative behaviour for individual agent ( $A_i$ )**)

Condition: The world state is changing continuously.

---> Behaviour: 1. *send-message(agent)*,  
2. *wait-for-message(agent)*  
3. *receive-message(agent)...*

Rule 6: (rule name: **attitude generator rule for medium fire**)

Condition: size of fire is less than two squares.

---> Action: generate attitude ***medium-fire***

Rule 7: (rule name: **attitude generator rule for dangerous situation**)

Condition: chemicals in the room.

---> Action: generate attitude ***dangerous-fire***

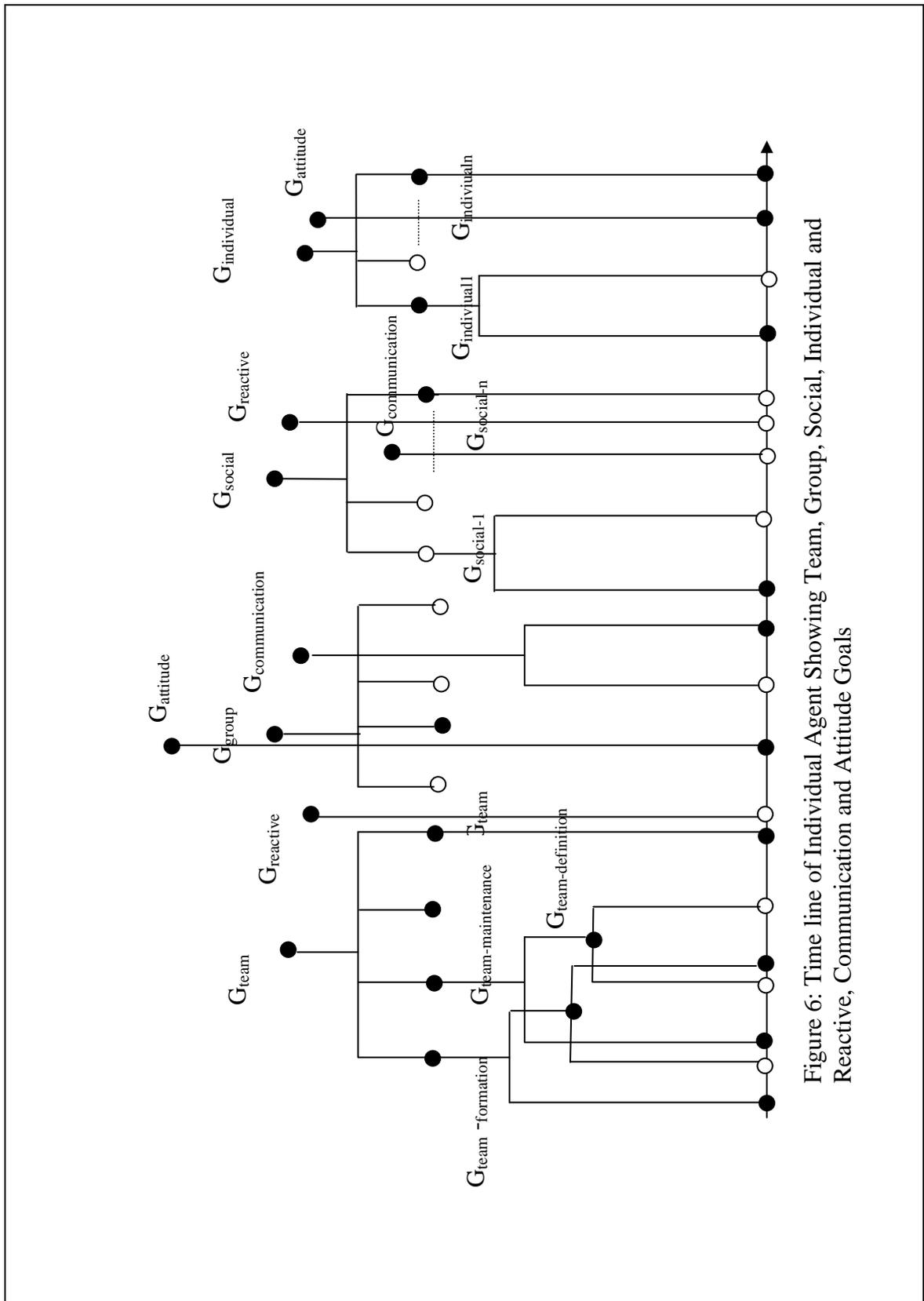


Figure 6: Time line of Individual Agent Showing Team, Group, Social, Individual and Reactive, Communication and Attitude Goals

## 5.1 Agent Architecture

We adopt the team agent's ( $A_T$ ) architecture for the individual agent ( $A_i$ ). Thus, the abstract behaviour of team agent and individual agent would appear similar. However, the attitudes and the goals would be different.

## 5.2 Execution of Individual Agent's ( $A_i$ ) Time lines

The team activities within the team must eventually rest on actions of individual members or subgroup. The collaborative team actions are usually highly abstract and consist of actions for individual members or subgroups. The time line structure for the team as well as of the agents is shown in figure 7. As described above, the team task is divided into the team goals  $g_1, g_2, \dots, g_n$ . Further team goal  $g_1$  is divided into individual actions of agents  $A_1$  and  $A_2$ .

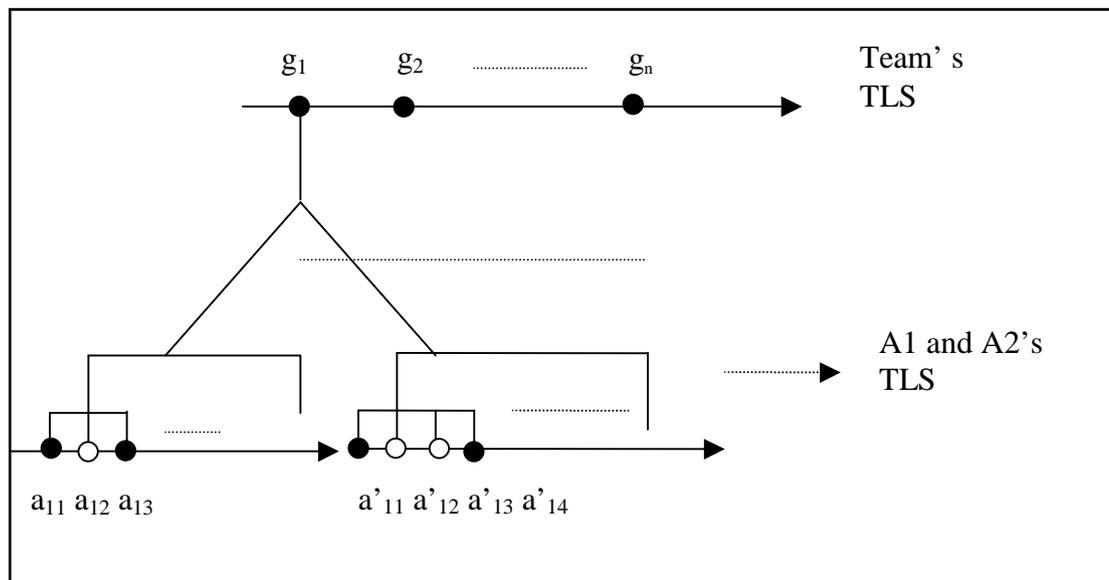


Figure 7: Time Line Structure of Two Agents

The time line of the individual agents usually contains the primitive actions, abstract actions, rules, conditional actions and loops. The execution of each of these actions is different from one another. The execution of time line structure (TLS) is a situation, in which a time line is being executed. The goal generator produces a large variety of goals depending upon the variations in the dynamic world. At any instant of time, the time line manager keeps inserting new nodes according to the priority of the different goals and actions. This problem solving strategy provides a framework that allows interleaving planning and execution. Thus replanning may be triggered in response to unexpected events. It means the agents will be able to update, modify and suspend their team plan if the situation in a dynamic environment changes. The following basic operations can be done on the time line structures:

1. Insert Goal.
2. Plan for Goal.
3. Execute Goal.
4. Insert attitude for a given node.
5. Wait for execution.
6. Continue execution.

7. Delay execution of Goal
8. Abandon Goal.
9. Able to mark the goal node according to the attributes.
10. At any instant of time able to display the statistics of the structure

## 6. An Example

In order to demonstrate the effectiveness of our attitude based model of team problem solving, we present a detailed example in this section. The example begins when the chair *ch\_1* in the cryogenic laboratory catches fire (Figure 8). When an agent senses the fire, it adopts the individual attitude *team-form*. When two agents ( $A_1$  and  $A_2$ ) adopt this attitude, they perform physical, mental and communicative actions to form a team structure and adopt an attitude *team* towards this structure. In the next step, the agents choose and adopt an attitude towards the fire. In this example, assuming that fire is of medium size, agents adopt the attitude *medium-fire* towards the fire. This attitude may be a *collective* attitude or *individual* attitude. To simplify the discussion, we assume that the attitudes are of *individual* type. The team rule base  $R^T$  will invoke team goal  $G_{(put-out-fire-campus)}$ . In order to put out the fire in the cryogenic lab, team agents have to move from open compound to the Cryogenic laboratory via Vet. Store Room.

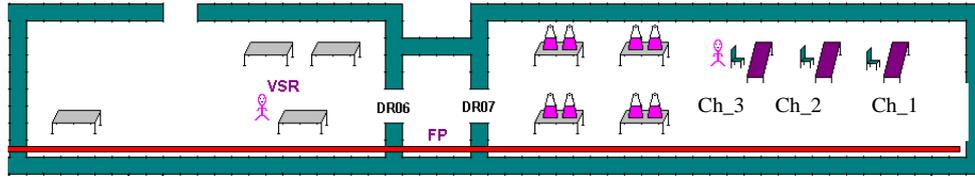


Figure 8: A Graphical Representation of the Section of the Domain

The situation in the Cryogenic Laboratory is also like that there are two victim agents ( $V_1$  and  $V_2$ ) in the room. Now the attitude *rescue* is generated and the abstract goal  $G_{(put-out-fire-campus)}$  generates two sub-goals:  $G_{(rescue-victims)}$  and  $G_{(put-out-fire-room)}$ . These two goals correspondingly generate team activities putting out the fire as well as rescuing the victims. In order to rescue the victims, the team will first put out the fire around the victims and then will ask them to come out of the Cryogenic Lab. The goal  $G_{(rescue-victims)}$  was then further decomposed into three sub-goals,  $g1_{(approach-victim)}$ ,  $g2_{(put-out-fire-victim)}$  and  $g3_{(victim-escape)}$ . The sub-goal  $g1_{(approach-victim)}$  results into deriving a team plan from team plan library. The derived plan outlines a path to get to a position in the Cryogenic lab. The plan is inserted into the TLS and the agents start moving by executing the move actions. The figure 9 shows that how the abstract goals  $G_{(rescue-victims)}$  and  $G_{(put-out-fire-room)}$  are executed on the individual agents  $A_1$  and  $A_2$ 's TLS.

To get to the veterinary storeroom from the cryogenic lab, agents have to get through two doors DR6 and DR7 (figure 8). The sub-goal  $g1$  is further decomposed into goals of agents  $A_1$  and  $A_2$ . The goal of agent  $A_1$  is decomposed into primitive actions  $a_{11}$ ,  $a_{12}$ , ...,  $a_{1n}$  which forms a path towards the victim. Similarly the goal  $a1'$  of agent  $A_2$  is decomposed into primitive actions  $a_{11}'$ ,  $a_{12}'$ , ...,  $a_{1n}'$ . The goal  $a_2$  is carried out followed by the completion of action  $a_{1n}$ . The  $a_{2(put-out-fire-victim)}$  is also a goal which involves putting out the fire around the victim agent. The abstract action  $a_{3(victim-escape)}$  is a conditional action asking the victim agent to escape, if there is no fire around the victim agent. If there is still fire around the victim, the team agents have to repeat the actions  $a_{2(put-out-fire-victim)}$  and  $a_{3(victim-escape)}$  i.e. there will be actions  $a_4$  and  $a_5$ .

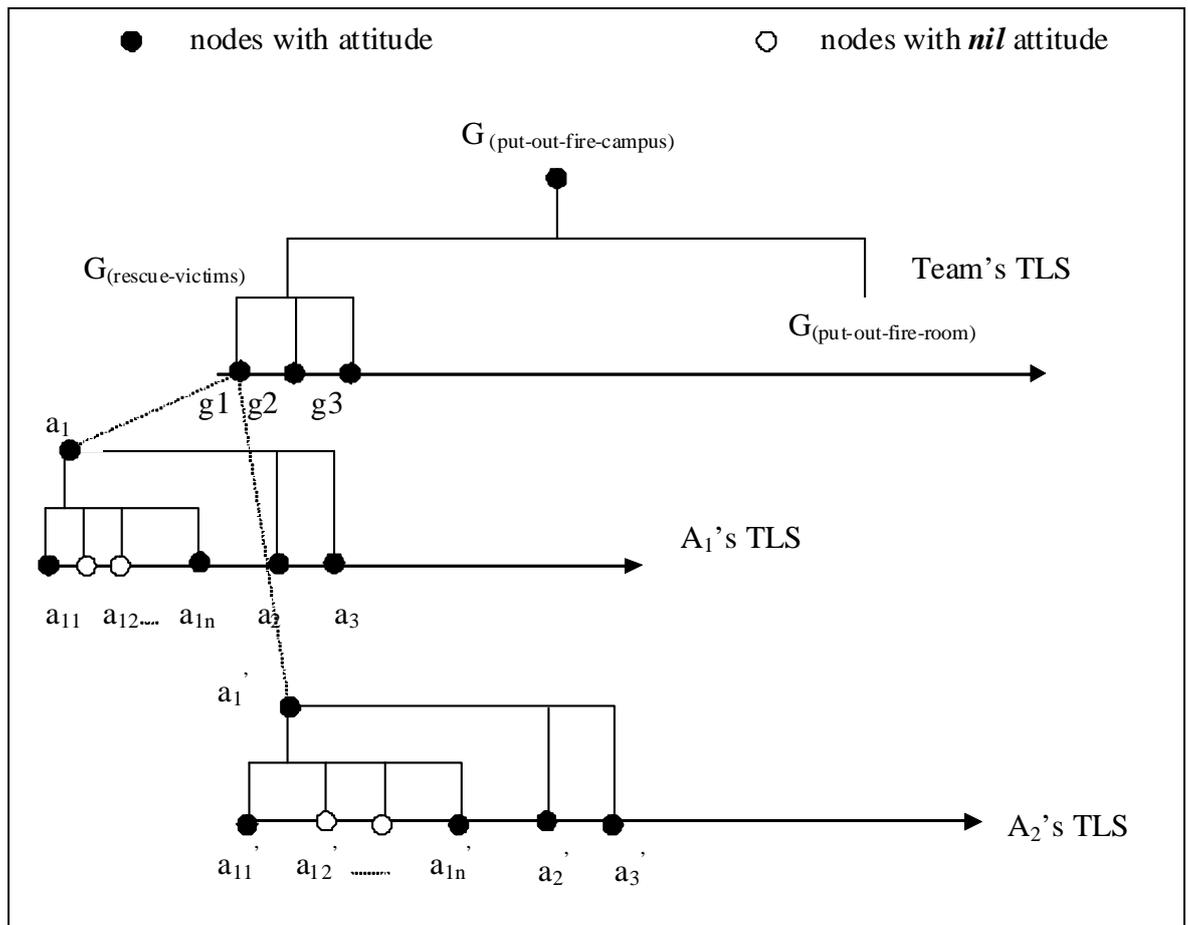


Figure 9: A1's, A2's and Team's TLS. Where  $a_1$  and  $a_1'$  -  $g1_{(approach-victim)}$ ,  $a_2$  and  $a_2'$  -  $g2_{(put-out fire-victim)}$  and  $a_3$  and  $a_3'$  -  $g3_{(victim-escape)}$ .

The above scenario describes the world when there are no changes in the world and everything goes according to the plan. But if there are changes in the world, there is change in the attitude of the agent. Due to the change in the attitude of the agent, new goals are inserted into the team plan. Suppose the fire occurs in the path of the firefighting agents while they are approaching the victim, so the firefighters have to find a new path for approaching the victim. Due to the changes in the state of the world, the agents adopt attitude *situation-team-maintenance*. This attitude results into insertion of team definition goal  $G_{(team-definition)}$  into the team plan. When the team definition goal is inserted into the team plan, the time line of the team as well as of the agents A1 and A2 is modified as shown in the figure 10. The actions (team definition)  $b_{11}, b_{12}$  and  $b_{13}$  are inserted into A1's time line and  $b'_{11}, b'_{12}$  are inserted into A2's time line.

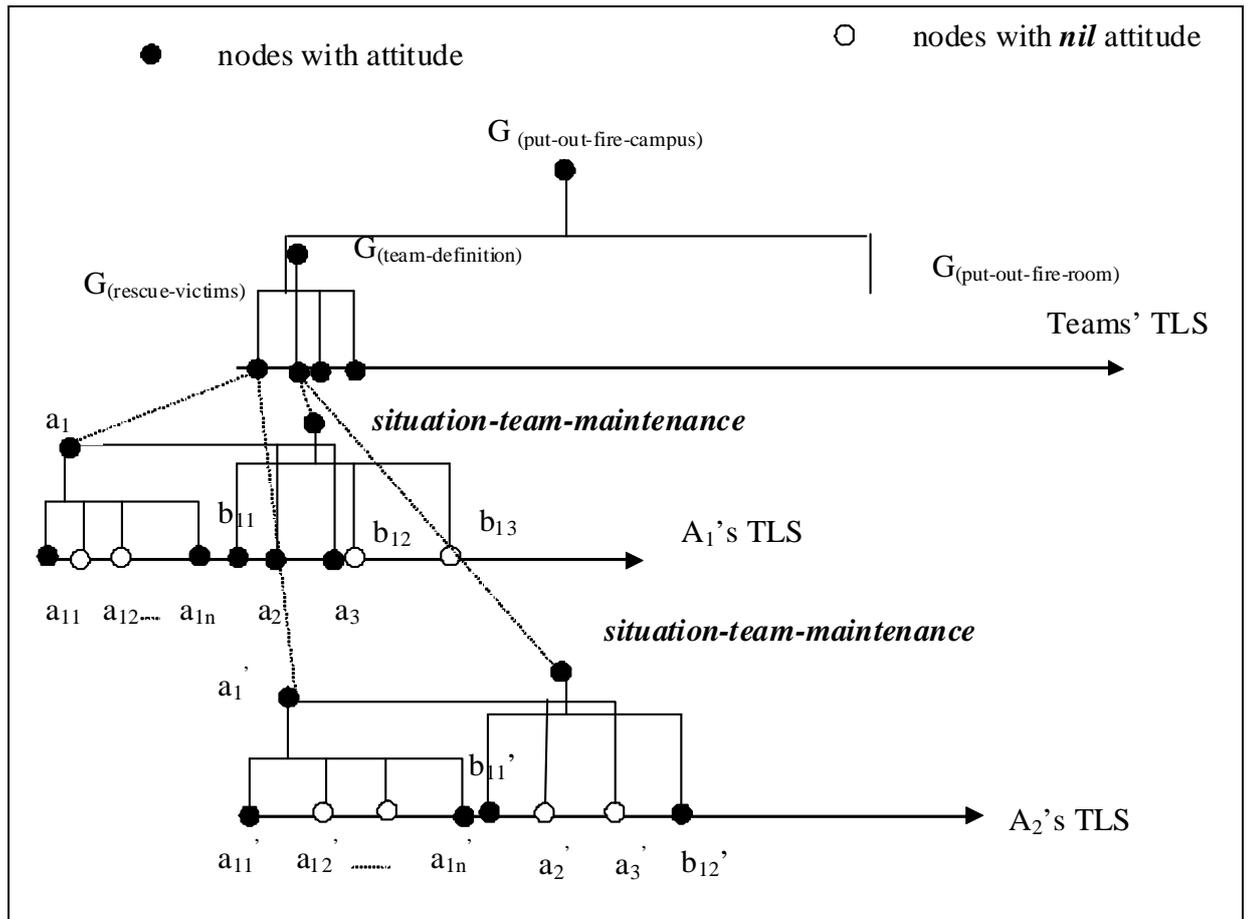


Figure 10:  $A_1$ 's,  $A_2$ 's and Team's TLS Showing the Effect of Attitude *situation-team-maintenance*(). Where  $a_1$  and  $a_1'$  -  $g_1$ (*approach-victim*) ,  $a_2$  and  $a_2'$  -  $g_2$ (*put-out fire-victim*) ,  $a_3$  and  $a_3'$  -  $g_3$ (*victim-escape*) and  $b_{11}, b_{12}, b_{13}, b_{11}', b_{13}'$  are team definition actions.

The changes in the world can be due to the changes in the fire amount or obstruction in the path of agent. If there is no obstacle in the way, the victims will come out of the room one by one. A conflicting situation arose when agent  $A_1$  was getting into the Cryogenic Lab., while agent  $V_2$  was coming out of the laboratory. Both agents needed to get through DR07, which was an essential resource in their respective plan. Multiple agents resolve the problem of conflicting resources by applying social convention goal  $G_{social-convention}$ . The social convention goal is invoked because of the social attitude *Wait*. Due to the social attitude *Wait*, the agent  $A_1$  will suspend its move action and wait for another agent  $V_2$  until it has finished moving away from the door. Therefore, a suspension action called *sus* and *wait* action are generated and inserted into  $A_1$ 's TLS. The social convention actions are completed when agent  $A_1$  recognises that victim  $V_2$  had passed through door DR07. The TLS of this time frame is shown in figure 11.

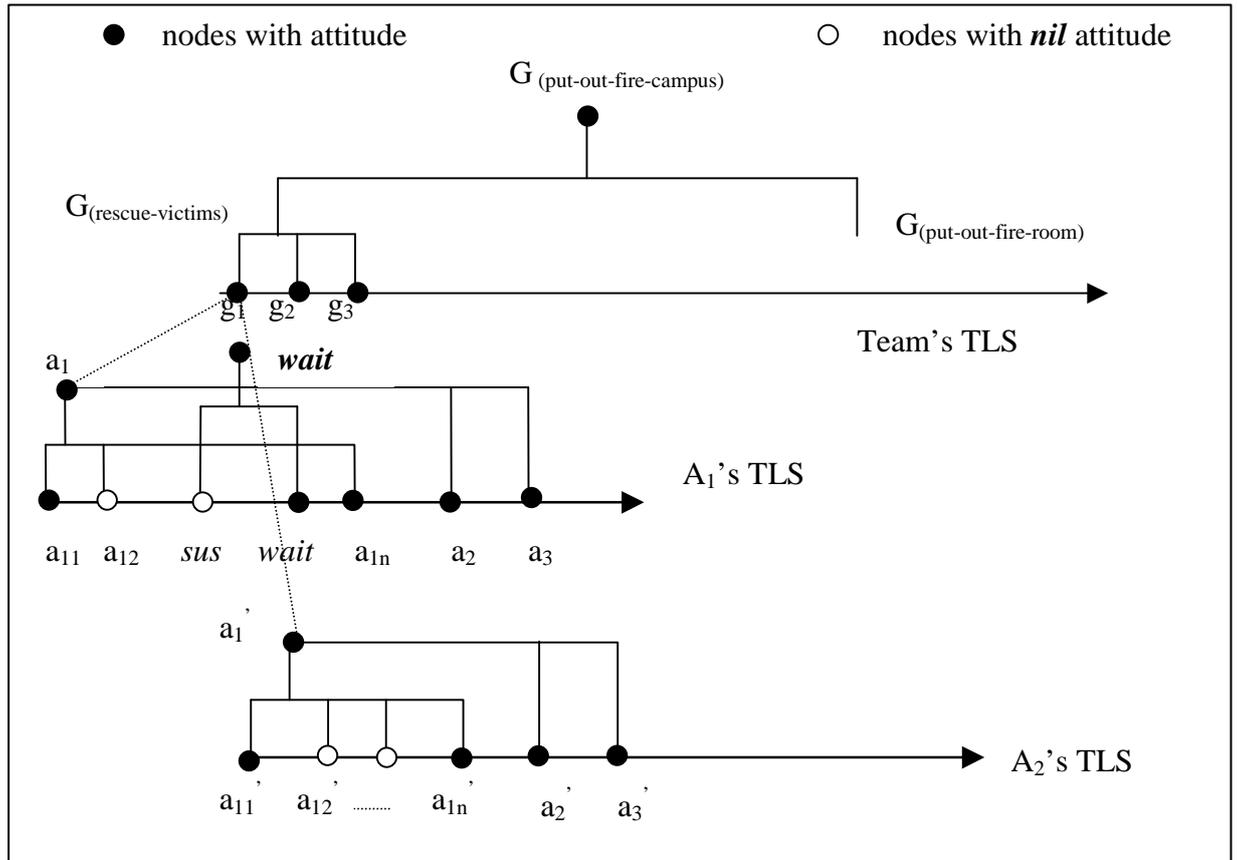


Figure 11: A1's, A2's and Team's TLS Showing the Effect of Attitude *wait()*. Where  $a_1$  and  $a_1'$  –  $g_1$ (approach-victim) ,  $a_2$  and  $a_2'$  –  $g_2$ (put-out fire-victim) ,  $a_3$  and  $a_3'$  –  $g_3$ (victim-escape) and *sus* and *wait* are social convention actions.

The agents in order to put out fire around the victim may have to move the chemicals out of the room. So the team goal generator will insert the goal  $G$  (move-chemicals) into the time line of the agents. The chemicals catch fire very quickly, so agents have to deal with them very carefully and cautiously. So the goal to move chemicals  $G$  (move-chemicals) is tagged with *cautious* and *warning* attitude. There were eight chemicals in the room namely  $C_1, C_2, C_3, \dots$  and  $C_8$  and consequently eight sub-goals are generated with the *full-coordination* attitude. The objective of the team is to move the chemicals to a safe location; in this case, the vet store room. Eight chemicals have to be removed from the cryogenic lab and placed on empty tables in the vet store room. So each sub-goal is further decomposed into the following sub-goals: *go-to-chemical*, *pick-up-chemical*, *move-chemical* and *put-down-chemical*. A graphical representation of the TLS structure showing *warning* and *cautious* attitude is given in figure 12.

While doing the team activity, the fire may become very large and out of control. So the attitudes *large-fire*, *escape*, *team-deform* are formed. A graphical representation of the TLS structure showing these attitudes is given in figure 13. By adopting these attitudes, the agents abandon the team activity and move towards the open safe place and this will conclude the team activity and our example. However, in this case if fire becomes less and is under control i.e medium fire, the fire-fighters will again form the team and start doing the task of putting out the fire.

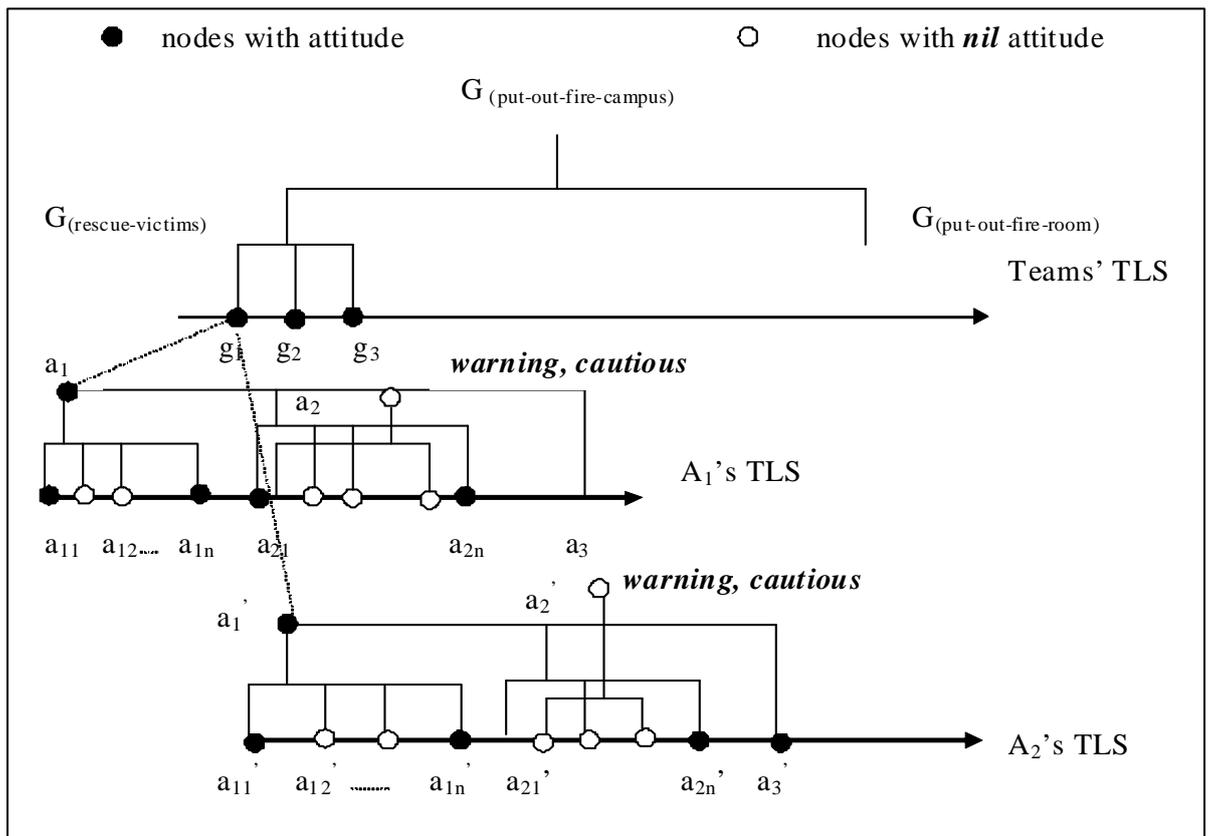


Figure 12:  $A_1$ 's,  $A_2$ 's and Team's TLS Showing the Effect of Attitudes *warning()* and *cautious()*. Where  $a_1$  and  $a_1'$  –  $g_1$ (*approach-victim*) ,  $a_2$  and  $a_2'$  –  $g_2$ (*put-out fire-victim*) ,  $a_3$  and  $a_3'$  –  $g_3$ (*victim-escape*)

In this section, we have examined the team problem solving behaviour of the agents with their accompanying attitudes through a detailed example. The events and behaviour of the agents can be summarised as follows:

1. Fire occurs in Cryogenic Laboratory on Chair  $ch\_1$ .
2. Agents  $A_1$  and  $A_2$  form team.
3.  $A_1$  and  $A_2$  go to the Cryogenic Laboratory.
4.  $A_1$  and  $A_2$  put out fire around victims  $V_1$  and  $V_2$ .
5.  $A_1$  gives way to  $V_2$  while passing through door DR07.
6.  $A_1$  and  $A_2$  move chemicals  $C_1$  and  $C_2$  to Veterinary Store Room.
7. Fire changes medium fire to large fire.
8. Agents  $A_1$  and  $A_2$  unform the team and escape.

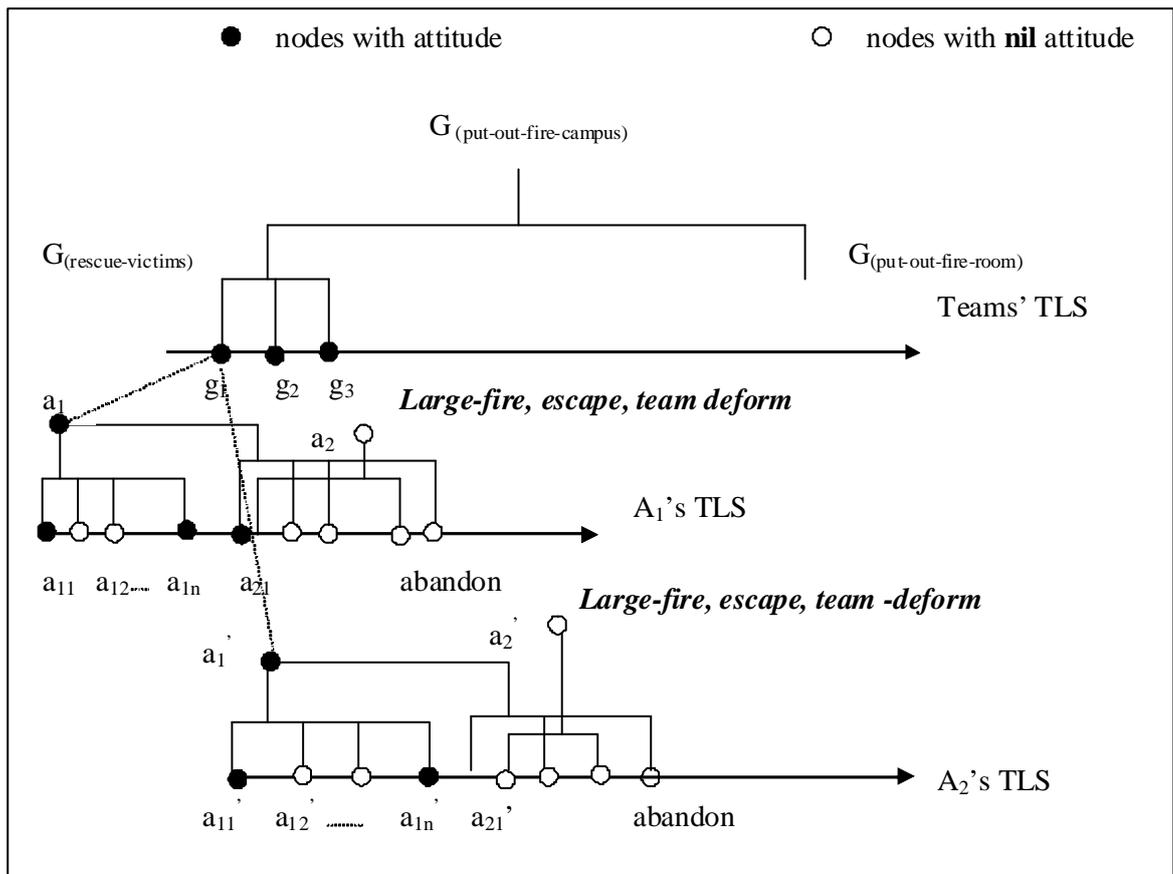


Figure 13: A<sub>1</sub>'s, A<sub>2</sub>'s and Team's TLS Showing the Effect of Attitudes *large-fire()*, *escape()*, *team-uniform()*. Where a<sub>1</sub> and a<sub>1</sub>' – g<sub>1</sub>(*approach-victim*) , a<sub>2</sub> and a<sub>2</sub>' – g<sub>2</sub>(*put-out fire-victim*) , a<sub>3</sub> and a<sub>3</sub>' - g<sub>3</sub>(*victim-escape*)

## 7. Conclusion

In this paper, the significance of various attitudes and their role in problem solving is demonstrated. A methodology of collaborative problem solving, which describes all the aspects of the problem solving in a fire world based on attitudes is presented. The overall team problem solving process is divided in two levels i.e problem solving by the team agent (A<sub>T</sub>) and problem solving by the individual agent (A<sub>i</sub>). In this paper, a common architecture for both team and individual agents is proposed to solve the team problems. To represent the problem solving execution details of team agent (A<sub>T</sub>) and individual agent (A<sub>i</sub>), we have attempted to show as much of details on time line structures (TLS) as possible. We have also given the methods of various team problem-solving activities in a Fire World. There are many useful pieces of work on collaborative agents like Plaides [11], ADEPT[8] and ARCHON[3]. Despite successful demonstrations of these projects, collaborative agents face the following problems.(i)Inter-agent coordination: coordination is essential to enable group of agents to solve problems effectively because of the constraints of resource boundedness and time. Without a clear theory of coordination, anarchy or deadlock can set in collaborative agents. (ii) Evaluation of collaborative agent systems – how they react to unanticipated events. However, attitude based collaborative agents don't face these type of problems.

Although we have not explicitly considered communication, our model is consistent with prevalent theories of *speech acts* [10]. In our model, the basic requirement of the communication is the capability to represent the *physical states* and *mental states*. In order to communicate about the world states, the agents are able to give description and put a query about the objects existing in this world and the relationship between these objects. The communication can be either about the location of objects or the state of objects. The state of objects in our domain can be normal, burning or burnt. To communicate mental states, agents are able to reply about goals, actions, individuals, group, society and team.

## References

- [1] I.Ajzen. Attitudes, Personality and Behaviour. Open University Press, Milton Keynes, England.
- [2] M.E.Bratman. Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, MA, 1987.
- [3]D. Cockburn and N. R. Jennings. ARCHON: A Distributed Artificial Intelligence System for Industrial Applications. In Foundations of Distributed Artificial Intelligence (eds. G. M. P. O'Hare and N. R. Jennings) Wiley, 1996, 319-344.
- [4] P.R.Cohen, M.L.Greenberg, D.M. Hart, and A.E. Howe. Trial by fire: Understanding the design requirements for agents in complex environments. AI Magazine, 10(3), 1989.
- [5] P.R.Cohen and H.J.Levesque. Teamwork. Special Issue on Cognitive Science and Artificial Intelligence, 25(4), 1991.
- [6] M.Fishbein and I.Ajzen. Belief, Attitude, Intention and Behaviour: An Introduction to theory and research. Reading, MA, USA: Addison-Wesley, 1975.
- [7]Marcus J. Huber, Edmund H. Durfee. On Acting Together: Without Communication. In Proceedings of AAAI Spring Symposium on Representing Mental States and Mechanisms, pages 60-71, Stanford, CA, March 1995, AAAI press.
- [8] N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, M. E. Wiegand, C. Voudouris, J. L.Alty, T. Miah, and E. H. Mamdani (1996). ADEPT: Managing Business Processes using Intelligent Agents. In Proceedings of BCS Expert Systems Conference (ISIP Track), Cambridge, UK 5-23, 1996.
- [9] H.Kitano, M.Tambe, P.Stone, M.Veloso, S.Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada. The RoboCup Synthetic Agent Challenge,97, International Joint Conference on Artificial Intelligence (IJCAI97).
- [10]Y. Shoham and M. Tannenholtz. On the Synthesis of Useful Social Laws for Artificial Agent Societies. In Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), pages 276-281-380, July 1992.
- [11]Katia P.Sycara, K. Decker, A. Pannu, M. Williamson and D. Zeng. Distributed Intelligent Agents. In IEEE Expert, Dec., 1996.
- [12] Milind Tambe. Agent Architectures for Flexible, Practical Teamwork. In Proceedings of the National Conference on Artificial Intelligence (AAAI), August 1997.
- [13]Mario Tokoro, An Agent Is an Individual That has Consciousness. In J.P.Muller, M.J. Woolridge and N.R. Jennings, editors, Intelligent Agents III – Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages (ATAL-96), Lecture Notes of Artificial Intelligence. Springer Verlag, 1996.
- [14]R. Toumella and K.Miller. We-intentions. Philosophical Studies, 53:367-389,1988.

[15]M. Wooldridge and N. R. Jennings. Towards a Theory of Cooperative Problem Solving. In Proceedings of the Sixth European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-94).