Chapter 3

TEAMWORK IN MOBILE AGENTS

3.1 Introduction

With the increase in complexity of applications and its distributed nature, the need of efficient and effective teamwork among software agents is becoming more critical. Teamwork among software agents can be accomplished using various combinations of team member patterns. However, mobile agents require special teamwork operation strategy due to its mobility and specific task oriented nature.

A teamwork strategy has been proposed and analyzed in context of mobile agents, where a group of software agents is tasked to perform a joint operation. The proposed technique matches with the Honey Bee teamwork pattern in real life analogies, so it is named as Honey Bee teamwork architecture. Two possible scenarios for teamwork have been used to explain the proposed architecture for teamwork among software agents.

3.2 Honey Bee and Mobile Agents

There are a number of similarities exist between working strategy of honey bees and mobile agents when they are performing their tasks in a group. The resemblance has been highlighted between working strategy of honey bees and mobile agents. It forms the basis of design for proposed teamwork architecture.
In the scenario of honey bees network, the queen bee leads the hive and manages as well as coordinates the major operations. All other bees provide various kinds of services to the queen. In mobile agents’ domain, team leader approach is proposed which manages, shares goal and plans as well as coordinates and monitors the operations of member agents. The team leader divides the top goal and plans the sub-goals for members and manages their operations and performance by information sharing, coordination and collaboration.

Honey bees roam around from flower to flower for fulfilling their major goal i.e. extracting nectar for making honey, which is later stored in the hive. Honey bees extract the nectar from many flowers as they fly. Bees retain it, return and release it at their hive at the designated cell. Similarly, mobile agents may visit various machines in order to fulfill their assigned tasks like searching for specific information. They extract the

---

**Figure 3.2 Classification of Teamwork Model**
required data and process accordingly. Mobile agents may keep the extracted information or manipulate at destination machine and deliver result after returning to their parent machines or hosts.

In case, the queen bee is dead, there exists a well defined process for finding and making a new queen. A special food is fed to a particular bee; this process brings up the new queen. The queen adopts the role of new leader of the hive. In the context of software agents, fault tolerance mechanisms have been introduced for generating new team leader in case of malfunctioning. A promotion algorithm has been used in which the member agent is elevated as team leader depending on its seniority of time scale.

3.3 Teamwork Architecture

The general teamwork architecture is classified as shown in Figure 3.1. The teamwork architecture may be divided into two main categories. Firstly team of members, which are working for the same prime goal but there is no direct collaboration and communication between them. In such type of teamwork, members are not aware about their peer or neighboring members. The only thing identical among them is their prime goal which is being pursued by each member.

Other category in teamwork shows working of member agents to fulfill primary goal under identical plan. Such members coordinate and communicate with each other as per requirement of goals and operations strategy. This strategy is also sub-divided in two
more categories. The first sub-category consists of team leader approach where a specific member is designated as leader and it is responsible for goal and plan division to its team members. In this approach, all information is shared through team leader among member agents of team. The second sub category is Non-team leader approach in which all members of team have direct communication with each other at peer to peer level without specifying a specific member as team leader.

![Figure 3.2 Interactions among Agents in Team Leader Approach](image)

The specialized team leader approach has been proposed for mobile agents as inferred from honey bee working strategy. A comparison of both approaches is discussed in later part of the section along with rational for assigning priority to team leader approach over non team leader strategy.

The major terms are defined as following,

Goal for team members – g

Task to be performed – tk

Team leader agent - T_l

Team member agents - T_m
Communication factor in team leader approach - $C_{TL}$

Communication factor in non-team leader approach – $C_{NTL}$

### 3.3.1 Team Leader Strategy

In the Team Leader scenario, there is one dedicated agent which is leader of the team and its responsibility is to allocate the goal or plan as well as tasks operation strategy to each member in the team as per specification. In this approach, collaboration among member agents occurs in a hierarchical fashion. It transpires that collaboration among member agents of specific team is taking place by way of team leader. Such technique allows convenient sharing and integration of critical information among members of team in mobile agents’ scenario. In this particular strategy, team leader agent shares the goal and tasks related information to its member agents in a tree like hierarchical structure as highlighted in Figure 3.2. The communication in this particular technique can be represented as $2n$ for $n>1$, where $n$ is the number of member agents in a specified team.

The elaborated flow of activities in this algorithm of teamwork is described in Figure 3.3. At the start of application, user or owner of agent specifies the number of agents to be created in a team. After specification of team structure and primary goal, system creates the team leader and member agents. Team leader adds the members into its team and sets the communication pattern. The team leader divides the primary goal and its associated tasks into sub-goals as well as sub-tasks depending upon the specific number of member agents in the team. Team leader assigns these sub-goals and sub-tasks to members and execution is started. In this approach, all communication takes place through team leader.
agent which is coordinating and sharing all concerned information with members. As

Figure 3.3 Team Leader Strategy
soon as the main goal is achieved and specified tasks are accomplished, team leader shares and synchronizes information with members and operation is concluded.

In domain of disaster management systems, the resource/target hunt scenario is used for proof of concept in this particular case as it is widely used example in the domain of teamwork. It is one of the major scenarios where robots teamwork strategies and their communication and coordination patterns are analyzed [81] by renowned research groups.

In our first example of target/resource hunt application scenario, the team leader agent and member agents are created by the system. Team leader agent assigns the prime goal data and area under consideration to each member agent for searching the required target location. In case of two agents in a team, the area under search is divided into two parts and allocated to team member agents. In case of four, eight or sixteen members in the specified team, the allotted area is further divided into respective parts and each member agent is assigned its area of search. The team leader agent adds the member agents in its team at the start of activity and later tracks member’s progress as well as path of locations. When the target/resource is discovered, the team leader agent informs all member agents in its team and the search operation is concluded.

A particular job execution in team leader scenario can be represented in tuple form as

\[ \{ T_l, T_m, g_n, tk_n, Cf_{TL} \} \].

The communication factor comparison is highlighted in evaluation section for further analysis and discussion about its effects on generic teamwork strategy.
3.3.2 Non Team-Leader Strategy

In Non Team Leader scenario, there is no team leader who divides the plan or set of actions for each member agent. The general plan or primary goal is shared by members at peer to peer level. The members work themselves according to main goal and plan in direct collaboration and coordination with each other.

In the Non Team Leader strategy, each agent in the team forms direct communication link with each other. In terms of implementation perspective, each member has two streams of communication; one is input and other output. The communication among agents in non team leader approach is much higher as compared with the team leader approach as shown in Figure 3.4, especially for team of two, three and four agents.

![Figure 3.4 Interactions among Agents in Non Team Leader Approach](image)

The flow of activities is also highlighted in Figure 3.5. In non-team leader approach, the system creates the team members at startup of application. Team members share the goals and tasks information with each other and synchronize their status. Members start

34
execution on defined tasks while sharing its progress and current situation with peer members. As soon as the primary goal is achieved, information is synchronized and operation is stopped.

![Flowchart of Non-Team Leader Strategy](image)

Figure 3.5 Non-Team Leader Strategy

Considering the proof of concept application of target hunt, the members are collaborating and sharing information with each other while searching the area for
target/resource. They start searching the area on their paths and at the same time communicating with each other about their current location. If two members come face to face with each other or come across in same path which is already traversed by other agent, they change their direction and move towards the un-explored paths. When one of the agents finds the target, all member agents are informed and search operation is stopped.

As the number of agents increases in a team, the communication also increases respectively. The communication relationship in this case can be represented as \( n(n-1) \), where \( n \) is the number of member agents in the specified team.

A particular job \( n \) execution in non-team leader scenario can be expressed in tuple form as:

\[ \{ T_m, g_n, t_k, C_f_{NTL} \} \]

### 3.4 Teamwork in Earthquake Management System (EMS)

Teamwork approaches have been expressed in multiple disaster management systems with focus on teams of rescuers in fire or explosion related disasters. We have used the example of earthquake management system to highlight the potential capabilities of software agents from start of emergency activities like sending alerts to monitor and manage the relief operations. This particular example is also discussed in upcoming chapters of thesis, highlighting the role of ontology based policies and formal modeling perspectives.
In EMS scenario as shown in fig 3.6, the main station and field stations are signifying the team leader strategy. As highlighted in the earlier example of team leader approach, the major information flow among member field stations is occurring through main station in order to minimize the communication overhead. In case of malfunctioning of the main station agent, one of the field stations is selected as leader station using the promotion policy. It is then used for information sharing, monitoring and management of tasks.

This EMS architecture design also forms one of the contributions and is discussed under major scenarios of proposed work. The major design features of Earthquake Management System are discussed in chapter 5 while the related implementation part is highlighted in chapter 6 under context of ontology based policies.