CHAPTER 4

FIPA SPECIFICATIONS

4.1 FIPA

FIPA stands for Foundation for Intelligent Physical Agents. It is a IEEE standard organization for agents and multi-agents which promotes agent based technology and the interoperability of its standards with other technology. FIPA started in 1996 as a Swiss based organization to produce software standards for heterogeneous and interactive agents and agent based systems. In March 2006 it unanimously became a part of IEEE organization. Now FIPA is moving one step further, in future the plan is to produce standards for agent technology to integrate and work with the non-agent technologies.

In 1997-98 FIPA released a set of specifications for interactive agents. This includes abstract architecture for agents, communication and message passing standards, agent management standards, agent services etc. Some of the main specifications are discussed below.

4.2 AGENT LIFE CYCLE

Agent life cycle as specified by FIPA is given in figure 4.1. Tiziana Trucco (2006). Agents goes through various stages and states throughout its life time, these are described by FIPA as follows Tiziana Trucco (2006).
- **Initiated**: The agent is created but it has not yet registered with AMS, hence has no name or address, so it can not communicate with other agents.

- **Active**: The agent is registered with AMS, hence has been given a name and address, so it can access all JADE features.

Figure 4.1 Agent life-cycle as defined by FIPA

- **Suspended**: Agent is stopped; no internal thread is being executed, so no agent behaviors are being executed.

- **Waiting**: Agents is blocked till some condition is satisfied, the internal thread is sleeping and will wake up when it is informed that the condition is satisfied, typically through a message.
• **Deleted:** Internal thread of agent is terminated, agent is definitely dead. Also it is unregistered from AMS.

• **Transit:** Agents enters this state when it is migrating from one location to another, messages for this agent during its migration phase are buffered at the old location and then sent to the new location.

### 4.3 ABSTRACT ARCHITECTURE

Agents can be written using variety of technologies. One of the goals of FIPA is to provide interoperability and reusability. When the agents are written using different technologies, it is necessary for interoperability to identify the common architectural parts of both and combine it under a abstract architecture which then can be formally related to every valid implementation. This leads to Architectural abstractions. Once the systems are described abstractly, it is easy to identify the fundamental elements and relationship between them. This relationship helps identify how interoperability can be achieved in these agent systems. Even though these systems have different implementations, the design is based on the common abstract architecture, which gives them the capacity to interoperate. But it still needs to be provided with the mechanism to interoperate with each other, which includes transport and encoding. For example one agent may transmit ACL message with OMG IIOP protocol whereas other agent system may be utilizing IBM’s messaging scheme. Hence primary goal of abstract agent architecture is to create a messaging protocol for communication between agents which may be using different transport, communication or messaging language.
Scope of this architecture includes:

1. A model of services available to agents and other services
2. Message transport interoperability
3. Supporting various forms of ACL (Agent Communication Language) representations
4. Supporting multiple directory services

FIPA describes the abstract architecture, this serves as base for the concrete architecture specifications. These specifications describe in precise details how to construct a agent system in terms of the programming language, protocols to be used, operating systems to be used etc. Figure 4.2 describes how one can go from abstract architecture to concrete architecture.

Figure 4.2 Abstract Architecture to Concrete Architecture
4.4 AGENT MANAGEMENT

Agent management provides a framework in which FIPA agents exit and operate. The reference model is established using this framework which provides base for creation, registration, location, communication, migration and retirement of an agent. The reference model is described in Figure 4.3.

Figure 4.3 Agent Management

- Agent: This is the computational process which represents the autonomous and communicating facility of any application.
- Directory Facilitator (DF): This is an optional component of Agent Platform (AP).
• It provides listing of services which are made available by different agents. An agent can register its services with Directory Facilitator

• Agent Management System (AMS): This is a mandatory component of AP. It provides control over use of and access to a agent platform. There is one AMS per AP.

• Message Transport Service (MTS) : This is the default communication method for agents on different APs.

• Agent Platform (AP) : This provides the physical infrastructure of the agent including the operating system, machines, DF, AMS, MTS and agents.

• Software: This consists of all non-agent executable collections of instructions accessible to agents.

4.5 THE AGENT BASED BIO-SIGNAL PROCESSING SCENARIO

The proposed work attempts to demonstrate the concept of developing Multi-Agent platform for processing of Bio-signals. It also demonstrates the concept of developing agents using JADE – Java Agent DEvelopment framework. The agents are trained, intelligent system that is capable of setting up the platform for processing the EEG / ECG / EMG waveforms. The agents themselves communicate with each other in decision making process.

The technical goal of this work is to develop a multi agent platform for processing of bio-signals aiming at assisting medical practitioners in developing standard examination procedures.
If a medical practitioner wants to have an expert opinion about EEG / ECG / EMG of his patient, Generic Agent can be invoked to which he has to specify the SSN (Social Security Number) of the patient, the type of the signal and the corresponding data file. The Generic Agent in turn will search for the Specific Agent – EEG Agent, ECG Agent, EMG Agent based on the signal type on the network and if found, the corresponding information will be passed to the specific agent by the Generic Agent. For example the EMG medical practitioner wishes to have an expert opinion, the EMG Agent with all necessary information, will look for an EMG Expert System (HINT, DARE, CANDID, MIYOSYS-II). Getting the Expert knowledge, the interpretation will be sent back to the Generic Agent through EMG Agent. The expert opinion will be displayed on the user side as well as it will be stored in Database by DB Agent for further references.