

# Modeling of Multi-Agent Oriented learning System for Impaired Students with JADE

**Shiladitya Pujari**

Department of Information Technology, University Institute of Technology, Burdwan University, Burdwan, West Bengal, India  
Email: shiladitya.pujari@gmail.com

**Sripati Mukhopadhyay**

Department of Computer Science, Burdwan University, Burdwan, West Bengal, India  
Email: dr.sripatim@gmail.com

**Abstract**— In this research paper we presented a model of multi-agent system based learning environment for physically impaired students. The learning system is modeled on the basis of both centralized as well as distributed multi-agent planning. The entire learning system keeps track of the type of impairment the student has and mode of interaction of the environment is set depending on the type of impairment(s). The system consists of agents which are developed using JADE agent technology that helps the students with disabilities to continue studies from their own places.

**Index Terms**— Agent, Disabled, Learning System, Impairment, Multi-agent System

## I. Introduction

In this research paper, modeling of a computerized home-based learning system for physically impaired students is described [1-3]. As the intelligent agent concept as well as multi-agent system is considered as the most cutting edge technology in the recent era, an automated learning system can be designed and modeled with the help of this technology. A multi-agent oriented distance learning system can be one of the best platforms for physically impaired students to carry out their studies. In our previous research papers we presented the basic model of this learning system which consists of seven modules. Each module is responsible for some goal specific tasks and is coordinated by some specific agents. Those agents work on the basis of distributed agent planning because tasks are accomplished through the communications and coordination between those agents. Centralized agent planning is also implemented as there is a central coordinator agent within the system. In this paper, we described about the types of impairments of the students, the mode of interaction between the students and the computer depending on the type of impairments and the entire learning process. We also described the roles and

responsibilities of all agents within the system and significance of developing of agents using JADE agent technology. FIPA agent platform as well as JADE architecture model is also discussed briefly.

## II. Agents and Multi-agent System (MAS)

Agent and multi agent system has many definitions. As per Michael Wooldridge [4], an agent is a computer system that is capable of independent action on behalf of its user or owner. In other words, an agent can figure out for itself what it needs to do in order to satisfy its design objectives, rather than having to be told explicitly what to do at any given moment. And a multi-agent system is one that consists of a number of agents, which interact with one another, typically by exchanging messages through some computer network infrastructure. In the most general case, the agents in a multi agent system will be representing or acting on behalf of users or owners with very different goals and motivations. In order to successfully interact, these agents will thus require the ability to cooperate, coordinate, and negotiate with each other, in much the same way that we cooperate, coordinate, and negotiate with other people in our everyday lives.

Multi agent systems may be considered as the latest software engineering paradigm in the recent era. This kind of systems may be used in those domains which consist of the following features:

- Knowledge is distributed in different locations.
- Several entities, while keeping their autonomous behavior, have to join their problem-solving abilities to be able to solve a complex problem.
- The problems in the domain may be decomposed in different sub-problems, even if they have some kind of inter-dependencies. [4]

If we analyze the learning system, we shall find out that the knowledge required for solving a problem is spatially distributed in different locations. It is obvious

that learning system for physically impaired students is a complex system and as there is no straightway software engineering standardization to solve this kind of complex system, a multi agent based system may be a better approach at that place.

### III. Types of Impairments and Modes of Interactions

It is important to define disability of a person properly and to identify the kind of disability also. World Health Organization (WHO in 1976) drew a clear three-fold distinction between impairment, disability and handicap.

- Impairment is any loss or abnormality of psychological, physiological or anatomical structure or function.
- Disability is defined as any restriction or lack (resulting from an impairment) of ability to perform an activity in the manner or within the range considered normal for a human being.
- Whereas, handicap is a disadvantage, for a given individual, resulting from impairment or a disability, which prevents the fulfillment of a role that is considered normal for that individual.

WHO reaffirmed this classification in 1980. In 2001, WHO issued the **International Classification of Functionality, Disability and Health (ICF)**. The ICF distinguishes between body functions (either psychological or physiological, such as vision) and body structures (anatomical parts such as eyes) in 2002. The ICF was endorsed officially by all 191 WHO Member States in the 54<sup>th</sup> Health Assembly on 22 May 2001.

- As per the Princeton University Press, disability is defined as the condition of being unable to perform as a consequence of physical or mental unfitness; "reading disability"; "hearing impairment" etc.
- The World Health Organization defines Disability as follows: "Disability is a complex phenomenon, reflecting an interaction between features of a person's body and features of the society in which he or she lives".
- A disability is a condition or function judged to be significantly impaired relative to the usual standard of an individual or group, defined by an organization named disabled-world.

Disabilities can be classified into some types and sub-types. The classification is as follows [5]:

Table 1: Classification of Disabilities

• Mobility and Physical Impairments
◦ Upper limb(s) disability
◦ Lower limb(s) disability
◦ Manual dexterity
◦ Disability in co-ordination with different organs of the body
• Spinal Cord Disability
• Head Injuries - Brain Disability
◦ Acquired Brain Injury (ABI)
◦ Traumatic Brain Injury (TBI)
• Vision Disability
• Hearing Disability
• Cognitive or Learning Disabilities
• Psychological Disorders

Among these types of disabilities or impairments, we emphasized on some of the most significant types of impairments and focused on those to model and design the learning system. At this stage, we focused mainly on some kind of mobility and physical impairments, such as upper limb (s) disability, lower limb (s) disability, vision disability or blindness, hearing disability and combination of those stated above. Modes of interaction between the computer system and impaired students is

registered and for every individual students and each time that mode of interaction is activated along with all the assistive tools when that particular student is entered into the system. After focusing on the most significant types of disabilities or impairments, modes of interaction are defined on the basis of those types of impairments which are shown below in table 2:

Table 2: Types of impairments and modes of interaction

Marking	Type of Impairment	Mode of Interaction	Assistive Tools
0000	Purely Visual Impairment	Vocie Recognition System (VRS), Brail Printing System, Press Button System	Head-phone, Microphone, Press-button device
0001	Purely Hearing Impairment	Normal Graphical Mode, Command-based Assistant	Common peripheral devices, Hearing Aids
0010	Upper Limb (hands) Impairment	Voice Recognition System (VRS), Command-based assistant, normal graphical mode	Head-phone, Microphone
0011	Lower Limb (legs) Impairment	Normal Graphical Mode	Common peripheral devices
0100	Visual + Hearing	Brail Printing and Brail input	Brail devices
0101	Visual + Upper Limb	VRS, Command-based assistant	Head-phone, Microphone
0110	Visual + Lower Limb	Same as purely visual impairment	Same as purely visual impairment
0111	Hearing + Upper Limb	Command-based assistant, normal graphical mode	Hearing Aids, common peripheral devices
1000	Hearing + Lower Limb	Same as purely hearing impairment	Same as purely hearing impairment
1001	Upper and Lower Limbs	normal graphical mode, VRS, Command-based assistant	common peripheral devices, Head-phone, Microphone

#### IV. Description of The Learning System

The learning system described in this paper is consisting of some modules. Each module is coordinated by different agents. Among these, some modules are working as per distributed agent planning, and some are working on centralized agent planning. Agents of each module (actually the agents within a specific module) can communicate between themselves for the purpose of achieving some common goal when necessary.

It is clearly depicted in Figure 1 that the proposed distance learning system is divided into several modules or sections. That means knowledge (whether knowledge is specific for a module or global for the entire system) is distributed in different sites, and different entities or modules has to communicate between themselves to solve a complex problem [4][5]. In this multi-agent system based learning system, different user interfaces are there for interaction between the students and the system or interaction between human experts with the system. In the consecutive text, each module of the system has been described briefly.

The learning system has several modules as stated above. Amongst those modules, the first one, i.e. the Assistive Tools Selection module comes with a user interface with the help of which, a student with any type of significant impairments stated earlier can register him/her with the learning system for an entire course. Through the user interface associated with the first module, a student has to give details about him/her, about the course the student like to enroll and some other inputs along with the type of impairment (s). The User Interface Agent (UIA) takes those input and stores

in its local database. UIA has the responsibility to analyze the input and to select the best mode of interaction for that student. UIA also generates a unique identification number for the student like the registration number of the students at Universities. Using this unique identification number, students are identified within the learning system throughout the entire course. This number holds information about the student, type of impairment and information about the mode of interaction fixed for that student. This unique id number is passed to different agents of different modules which is described later. Each time the student who is entering into the learning system has to give the unique id number as input and the best mode of interaction will be activated along with all the assistive tools necessary. The system also keeps track of the activities of the students [8].

From Table 2, it can be said that ten types of impairment or combination of impairments has been identified with significance and the learning system may deal with. These types are marked from 0000 to 1010 (decimal 0 to 9). This information is embedded into the unique id number. Again from the table 2, we can see that at least five kinds of modes of interaction can be defined, i.e. Voice recognition system (VRS), Brail printing system, Press-button system, Command-based assistant and normal graphical mode. These are accordingly marked as 000 to 100 (0 to 4) for five modes. The unique id number holds this information also.

Here, figure 1, is the complete schematic diagram of the multi-agent oriented learning system for physically impaired students. From this diagram, we can see that, a student first selects an assistive tool or a set of assistive

tools required for him/her to carry out the learning process. Students use an interactive user interface to perform the entire necessary task relevant to the learning process. The user interface is coordinated by

the User Interface Agent (UIA). The responsibilities and roles of every modules as well as agents have been described in the following sections of this paper.

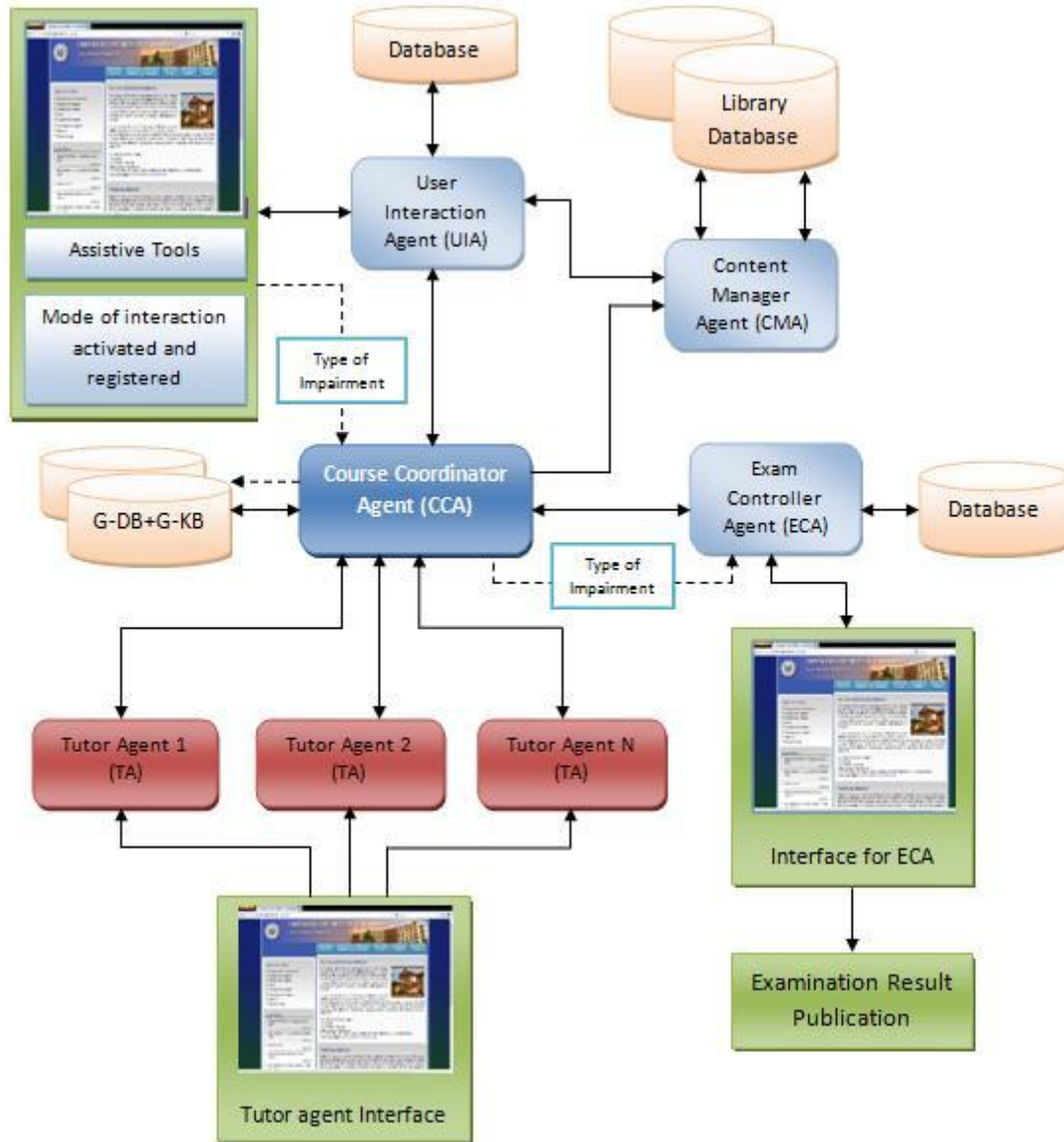


Fig. 1: Agents and their role in this system

After registration and selection of mode of interaction, the interactive user interface will help the student to start and carry out the learning process. Here, User Interface Agent (UIA) is entirely responsible for these kinds of interactivities and coordination as stated earlier. The UIA is associated with its own database and knowledgebase to keep the user information and serve the user queries properly in such a manner that it can interact with the user like human assistant. UIA receives input from user, does some analytical operation to make the input into knowledge, and then communicate with agents of other modules such as Course Coordinator Agent (CCA), Content Manager Agent (CMA) or Exam Controller Agent (ECA) etc. whenever necessary.

Throughout the entire course, UIA actually helps the user to communicate with Tutor Agents (TA) for different subjects to continue the regular learning and tutorial processes. The learning process is then carried out through the interactive user interface in proper mode of interaction. Each subjective Tutor Agent (TA) has its own local database to store tutorials and books, and has its own local knowledgebase also in order to interact and communicate with user as well as other agents within the system. The unique identification number (u\_ID) for every student is sent to each TA to keep track of their activities. TA for a specific subject helps the student to carry out their studies and it serves the subject related queries. As the u\_ID is sent to all

TAs, activation of proper mode of interaction is done while learning process is going on through the user interface of each TA. Using the interface in proper mode of interaction, a student can carry out subjective studies chapter wise as per the class schedule given earlier during registration process.

The Exam Controller Agent (ECA) is here for examination purposes. It has the responsibilities to conduct examination at proper time as per schedule. As the information and `u_ID` of the student is passed to the Course Coordinator Agent (CCA) and CCA keeps those data in the global database and global knowledgebase associated with it, each agent within the system can keep track of the entire activities of the student. With the help of CCA, exam controller agent (ECA) can fix the examination schedule. All the TAs for each subject keeps the question papers as per the examination schedule. ECA gives an alert message to the student about the examination and fixes the date/time for the examination. A student can take the examination through the user interface of ECA and can submit their answer script. Multiple choice answer script can be evaluated within the system with the help of TAs, but evaluation of descriptive type answer is done by human experts through the user interface of ECA. ECA also has the responsibility of tabulation and publication of results.

Content Manager Agent (CMA) maintains the central digital library of the system. Students can directly access the central library database for most of the time with the help of CMA. In some goal specific situation, Each TA communicates with CMA through Course Coordinator Agent (CCA), especially when the queries given by the student cannot be served by the TA itself. CMA also has the responsibility to store set of question paper and answer script for every student as well as the results for future references.

As we stated earlier, agents of this learning system work on the basis of both centralized as well as distributed multi-agent planning for communication, cooperation and negotiation between themselves [4]. The Course Coordinator Agent (CCA) acts as the major part of centralized multi-agent planning. CCA communicate with agents of every other modules of the distance learning system. It communicates with User Interface Agent (UIA), Exam Controller Agent (ECA), Content Manger Agent (CMA), and all the expert agents of the tutorial and learning section. Actually, each agent communicates and negotiates with other agents in the system through the CCA. That means CCA acts as the central coordinator. CCA refines and rectifies communications between agents if there are any flaws. Every agent is associated with their respective local database and knowledgebase, and they can also access the global database and global knowledgebase, through the CCA, if required. CCA directly accesses the global database and global knowledgebase of the system. CCA also keep track of

all the subject wise expert agents (Tutor Agent or TA) and their activities [5].

## V. FIPA Agent Platform and JADE Architecture Model

### A. FIPA Agent Platform:

Discussion on FIPA is necessary because JADE is largely an implementation of the FIPA specifications and it is highly dependent on the ideas generated through the specification process and expressed in the documents themselves. Moreover JADE has extended the FIPA model in several areas, the specifications do not provide complete coverage. However, JADE is compliant in all aspects relating to interoperability, which is the core purpose of FIPA.

Throughout the course of evolution of FIPA several agent-related ideas have been proposed. Among these ideas, the most important to the work are agent communication, agent management and agent architecture [6].

#### • Agent Communication:

The FIPA-ACL is grounded in speech act theory which states that messages represent actions or communicative acts which is called as speech acts or performatives. The FIPA-ACL is a set of 22 communicative acts and based on the ARCOL proposal of France Telecom where every act is described using both a narrative form and a formal semantics based on modal logic (Garson, 1984) that specifies the effects of sending the message on the mental attitudes of the sender and receiver agents. This form of logic is consistent with the BDI – or Belief, Desires, Intention reasoning model (Rao and Georgeff, 1995). Some of the most commonly used acts are *inform*, *request*, *agree*, *not understood*, and *refuse*. FIPA standards state that a fully compliant agent must be able to receive any FIPA-ACL communicative act message and at the very least respond with a *not-understood* message if the received message cannot be processed. Based on these communicative acts, FIPA has defined a set of interaction protocols, each consisting of a sequence of communicative acts to coordinate multi-message actions.

#### • Agent Management:

The second fundamental aspect of agent systems addressed by the early FIPA specifications is agent management: a normative framework within which FIPA compliant agents can exist, operate and be managed. It establishes the logical reference model for the creation, registration, location, communication, migration and operation of agents.

**Agent Platform (AP)** provides the physical infrastructure in which agents are actually deployed.

The AP consists of the machines, operating systems, FIPA agent management components, the agents themselves and any additional support software.

**Agent** is a computational process that inhabits an AP and typically offers one or more computational services that can be published as a service description. The particular design of these services, otherwise known as capabilities, is not the concern of FIPA, which only mandates the structure and encoding of messages used to exchange information between agents (and other third party technologies if FIPA compliant). An agent must have at least one owner and must support at least

one notion of identity which can be described using the FIPA Agent Identifier (AID) that labels an agent so that it may be distinguished unambiguously. An agent may be registered at a number of transport addresses at which it can be contacted.

**Directory Facilitator (DF)** is an optional component of an AP providing yellow pages services to other agents. It maintains an accurate, complete and timely list of agents and must provide the most current information about agents in its directory on a non-discriminatory basis to all authorized agents.

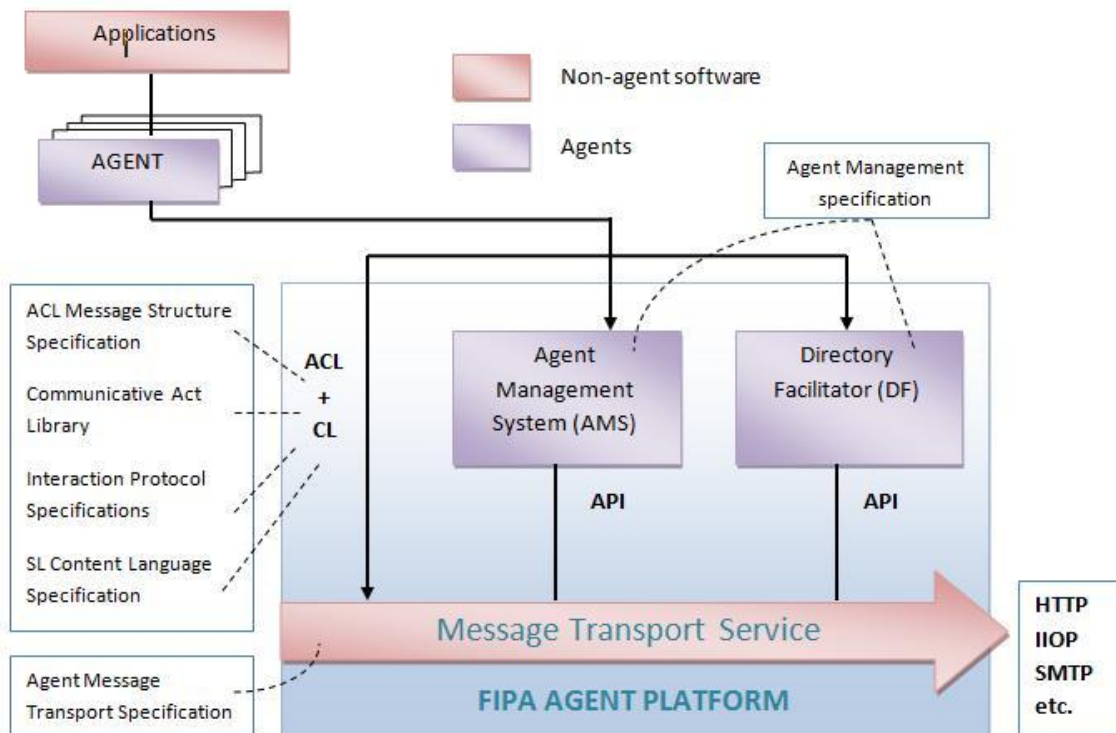


Fig. 2: FIPA Agent Platform

- **Agent Management System (AMS):**

The AMS is a mandatory component of an AP and is responsible for managing the operation of an AP, such as the creation and deletion of agents, and monitoring the movement of agents to and from the AP. Each agent must be registered with an AMS in order to obtain an AID which is then retained by the AMS as a directory of all agents present within the AP and their current state (e.g. active, suspended or waiting). Agent descriptions can be later modified under restriction of authorization by the AMS. The life of an agent with an AP terminates if it *deregister* itself from the AMS. After deregistration, the AID of that agent can be removed from the directory and that AID can be made available to other agents who should request it. Agent descriptions can also be *searched* for within the AMS, and the AMS keeps the AP description that can be retrieved by requesting the action *get-description*. The AMS can request that an agent performs a specific management function, such as to terminate its execution,

and has the authority to enforce the operation if the request is ignored. Only a single AMS can exist in each AP and if the AP spans multiple machines, the AMS is the authority across all those machines.

- **Message Transport System (MTS):**

The MTS is a service provided by an AP to transport FIPA-ACL messages between agents on any given AP and between agents on different APs. FIPA agent platform [6] is shown in Figure 3.

## B. JADE Architecture Model:

JADE is a middleware for the development of distributed multi-agent applications. According to the multi-agent systems approach, an application based on the JADE platform is composed of a set of cooperating agents, which can communicate with each other through message exchange. Each agent is immersed within an environment that can be acted upon and from whom

events can be perceived. The environment is dynamic in nature and agents created and terminated in the system according to the needs and the requirements of the applications. JADE provides the basic services necessary for distributed peer-to-peer applications in the fixed and mobile environment which allows each agent to discover others dynamically and to communicate with them. From the application point of view, each agent is identified by a unique name and provides a set of services. It can register and modify its services and/or search for agents providing given services, it can control its life cycle and, in particular, communicate with all other peer agents by exchanging asynchronous messages. Communication between the agents, regardless of whether they are running in the wireless or in the wire line network, is completely symmetric with each agent being able to both initiate an interaction and respond to it. JADE is fully developed in Java and is based on the following principles [7]:

- *Interoperability.* JADE is compliant with the FIPA specifications. As a consequence, JADE agents can interoperate with other agents, provided that they comply with the same standard.
- *Portability.* JADE provides a homogeneous set of APIs that are independent from the underlying network and Java edition. More in details, the JADE run-time provides the same API for the J2EE, J2SE and J2ME environment and it has been

designed and optimized for low footprint and memory requirements.

- *Ease of use and faster time-to-market.* The set of APIs has been designed with the goal of reducing the time to market for developing applications; therefore, they aim to hide the complexity of the middleware behind a simple and intuitive set of APIs.
- *Pay-as-you-go philosophy.* Programmers do not need to use all the features provided by the middleware. Features that are not used do not require programmers to know anything about them neither they add computational overhead.

### C. The Architecture:

JADE includes both the libraries (i.e. the Java classes) required to develop application agents and the run-time environment that provides the basic platform-level services and that must be active on the device before agents can be executed. Each instance of the JADE run-time is called *container* (since it “contains” agents). The set of all containers is called *platform* and provides a homogeneous layer that hides from agents the complexity and the diversity of the underlying tiers (hardware, operating systems, types of network, JVM) as depicted in Figure 3.

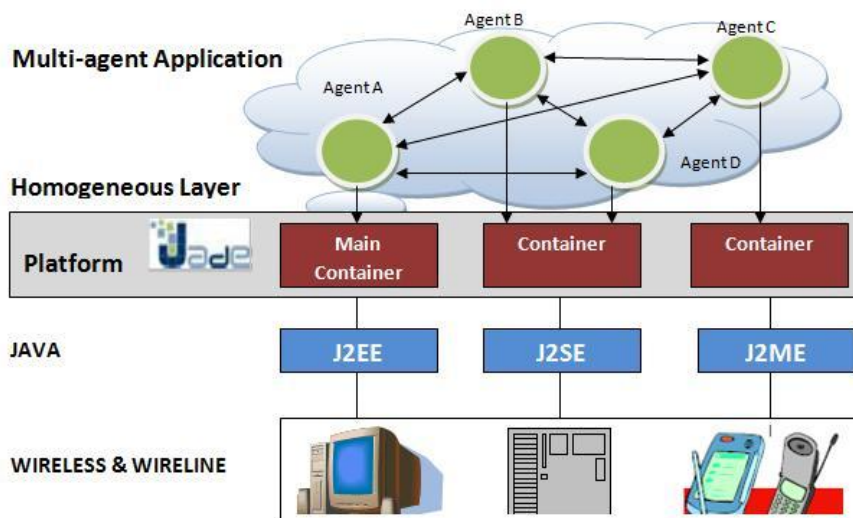


Fig. 3: JADE Architecture Model

For bootstrapping and FIPA compliance purposes, one among these containers is labelled as *Main Container*; it must be the first to start up and all other containers register with it at bootstrap time. The Main Container has the following specific responsibilities [6][7]:

- It manages the *Container Table* (i.e. the set of all the nodes that compose the distributed platform).

- It manages the *Global Agent Descriptor Table* (i.e. the set of all the agents hosted by the distributed platform, together with their current location).
- It manages the *Message Transport Protocols Table* (i.e. the set of all deployed message transport endpoints, together with their deployment location).
- It hosts the platform *Agent Management Service (AMS)* agent, mandated by FIPA specifications as unique white page and life-cycle management agent.

- It hosts the platform *Default Directory Facilitator (DF)* agent, mandated by FIPA specifications as default yellow page management agent.

All the above operations are essential for correct and FIPA-compliant platform operation. Unfortunately, this also entails that, from a fault tolerance perspective, the Main Container is a sensible part of the platform, and a single point of failure for many tasks. For this purpose, the replication service, described in the followings, allows to monitor and replicate the main container responsibilities. Its distributed architecture allows deploying JADE platforms that run across multiple Java editions, from powerful servers to small mobile devices. The limited memory footprint, in fact, allows installing JADE on all mobile phones provided that they are Java-enabled. JADE is compatible with the J2ME CLDC/MIDP environment and it has already been extensively used on the field over GPRS network with several commercial mobile terminals. The JADE runtime memory footprint, in a MIDP1.0 environment, is around 45 KB, but can be further reduced until 20 KB using the ROMizing technique, i.e. compiling JADE together with the JVM. JADE is extremely versatile and therefore, not only it fits the constraints of environments with limited resources, but it has already been integrated into complex architectures such as .NET or J2EE where JADE becomes a service to execute multi-party proactive applications [7].

## VI. The Implementation Scheme and Software Architecture of The Learning System

Initially the user (student) gives the essential information as input for registering him/her to the learning system. After the registration, a unique identification number (u\_ID) is generated for that student as stated earlier. Appropriate mode of interaction is also registered. After this process, user is authenticated and he/she can select the desired course of study from through the user interface. UIA takes the input and requests from the student. J2EE parser creates a XML file from the input queries. UIA stores those files into its local database and passes those files as well as the u\_ID of the student to the CCA. The central agent CCA analyzes the input and as per request, allocates subjects to the course of the student along with TA. CCA passes the u\_ID of the student to the TAs and to ECA and CMA. The XML file is parsed by the J2EE parser embedded in the user interfaces and data is transferred to the agents within the system. During the continuation of the course, student uses the TA interface to communicate with subject-wise TA. J2EE parser of this interface creates XML files and parsed it for communication purpose. Similar steps occurred when the student uses ECA interface and CMA interface of the ECA and CMA agents respectively. Figure 4 clearly showing the interaction and communication between the agents and user.

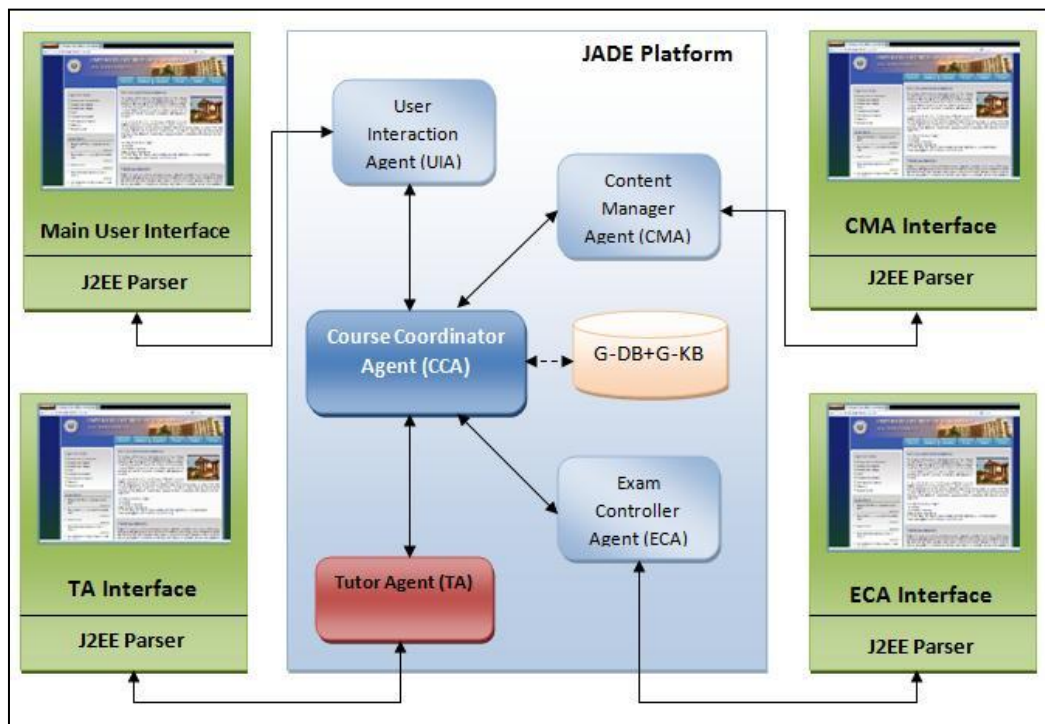


Fig. 4: Design Architecture of the Learning System

One of the key components of multi-agent systems is agent communication. Agents within a system practically need to be able to communicate with users, with system resources, and with each other if they need

to cooperate, collaborate, and negotiate. In particular, agents interact with each other by using some special communication languages, called agent communication languages (ACL) and that provide a separation between



the communicative acts and the content language. Inter-agent communication must be systematically analyzed. In this purpose, petri-net is one of the useful tools for modeling and analyze multi-agent system based applications [9].

Currently the most used and studied agent communication language is the FIPA ACL, which incorporates many aspects of KQML (Labrou et al., 1999). The primary features of FIPA ACL are the possibility of using different content languages and the management of conversations through predefined interaction protocols. Coordination is a process in which agents engage to help ensure that a community of individual agents acts in a coherent manner (Nwana et al., 1996). There are several reasons why multiple agents need to be coordinated including:

- agents' goals may cause conflicts among agents' actions,
- agents' goals may be interdependent,
- agents may have different capabilities and different knowledge, and
- Agents' goals may be more rapidly achieved if different agents work on each of them. Coordination among agents can be handled with a variety of approaches including organizational structuring, contracting, multi-agent planning and negotiation [7].

## VII. Conclusion

Here is the brief description of the proposed learning system for physically impaired. This learning system is a multi-agent based system, and this system can be undoubtedly a better alternative for the physically impaired students to continue their studies from their won locations. This system can also be accessed from home through the Internet if necessary. That means, a student with disability does not have to go to any institution for his/her study. This aspect may hopefully increase the number of literate disabled students in our country.

## References

- [1] Syed Asif Ali, S. M Aqi/ Burney, "Conversion of Heterogeneous Education System (HeES) into Homogeneous Education System (HoES) for Ease of Disabled Persons using Information Technology", International Conference On Computer Design And Applications (ICDA 2010), ISBN: 978-1-4244-7 164-5, 2010 IEEE.
- [2] Yin Bing, Lin Zhuying, Du Li, "The Applications of Multi-Agent Technology in Distance Education System", International Conference on Networking

and Digital Society, ISBN: 978-1-4244-5161-6/10 ©2010 IEEE.

- [3] WANG Dong-sheng, PAN Wei-wei, "The Design of Distance Learning System based on Multi-Agent", Second International Workshop on Education Technology and Computer Science (ETCS), ISBN: 978-0-7695-3987-4/10 © 2010 IEEE.
- [4] Michael Wooldridge, "An Introduction to Multi agent Systems", Department of Computer Science, University of Liverpool, Uk, JOHN WILEY & SONS, LTD.
- [5] S. Pujari, S. Mukhopadhyay, "Agent Oriented E-learning System for Visually Impaired Students using JADE Agent Technology", International Journal of Advanced Research in Computer Science, 3 (3), May –June, 2012.
- [6] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, "Developing Multi-agent System with JADE", JOHN WILEY & SONS, LTD, ISBN: 978-0-470-05747-6.
- [7] Fabio Bellifemine, Giovanni Caire, Giosuè Vitaglione, Giovanni Rimassa and, Dominic Greenwood, "The JADE Platform and Experiences with Mobile MAS Applications", ISBN 3-7643-7347-4 Birkhäuser Verlag, Basel – Boston – Berlin.
- [8] S. Pujari, S. Gupta, S. Mukherjee "A Multi-agent System (MAS) Based Scheme of Assistive Learning System For Handicapped and Disable Students". ISBN 978-93-80813-18-9, 2012.
- [9] S. Pujari, S. Mukhopadhyay "Petri-net: A Tool for Modeling and Analyze Multi-agent Oriented System". I.J. Intelligent Systems and Application (IJISA), 4(10), pp 103-112, August 2012.

**Pujari Shiladitya**, ME, PhD (Cont.) is an assistant professor in the department of Information Technology of University Institute of Technology, Burdwan University, India. He is working in the field of Multi-agent System, Artificial Intelligence as well as Cryptography and Steganography.

**Mukhopadhyay Sripati**, PhD, is working as a senior professor and head, department of Computer Science, Burdwan University, India. He is also acting as the registrar of the university for time being. He is working in the field of artificial intelligence for more than 15 years