Formal Verification, Architecture, and Implementation of User Location Guidance System Based on Multi-agents: A Case Study

Hina Faryal
MSCS
The Islamia University Bahawalpur, Pakistan
faryal.virgo@gmail.com

ABSTRACT: *A multi-agent system is used to solve problem that is difficult or impossible to solve by centralized system. Our multi-agent system provides a system that is formally verified. We have analyzed the development process of a multi-agent system after classifying it in the major phases of formal verification specifications, architecture specifications and implementation. Formal verification helps confirm that your software models and code behave correctly. We are using Formal model-checking approaches for the architecture and verification specifications.*

Formal verification, architecture, and implementation of an excursionist multi-agent system are implemented by using JADE (Java Agent Development Framework). JADE is a software framework fully implemented in Java language. It simplifies the implementation of multiagent systems with FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases.

This multi-agent system informs a user about their exact location, and guides him to find other locations (i.e. faculties, offices, hostels, banks, other buildings etc) inside the Bagdad-ul-Jaded Campus. Our system is based on Cartesian coordinates. Depending on the type of problem under consideration, coordinate systems possessing special properties may allow particularly simple solution. In addition to locating the position of user it also locates and guides him towards the other locations.

Keywords: Multi-Agent System (MAS), Formal Verifications, Architecture Specifications, Formal Methods, Formal Architecture, Agent Communication Language (ACL), Foundation of Intelligent Physical Agents (FIPA), JADE (Java Agent Development Environment). Finite State Process (FSP), Labeled Transition System (LTS)

Received: 19 January 2015, Revised 12 March 2015, Accepted 17 March 2015

1. Introduction

When we use an agent approach, sometimes problems required one or more agents. The distributed nature of multi agent approach solves the problems of multiple logics, and multiple interests. In addition, these factors interact with each other to achieve their personal goals and to manage dependencies. These interactions shared environment can change semantic interoperability with a simple type of server / client interaction traditionally with the ability to negotiate cooperation, coordination and action plan.

Agent is a computer system that is capable of independent action (standalone), on behalf of the user or owner (understand what needs to be done to meet the design goals, and as long as he told me).Agents do things that act: that is why they are called agents [Shardlow, 1990].

Autonomous agents and multi-agent systems represent a new way for complex system software to analyze, design and implementation of a new system. The view of the agent-based provides a powerful repertoire of technical tools and metaphors that have the potential to significantly improve the way in which people perceive and implement many types of software [Jennings et al., 1998].

Multi-Agent System was used in many areas to improve efficiency and quality, and reduce costs and save time by using the tools available to solve many problems. For complex systems, interacting autonomous agents offer a promising software engineering approach for developing applications. However, multi-agent systems launch a number of new abstractions and design / development compared to traditional approaches to software development. Therefore requires new methods of analysis and design and new tools to effectively build these systems.

Formal tests and methods, these two approaches can be used in developing a high quality software. While traditionally these approaches have been seen as rivals, in recent years, a new consensus is developed, which is used as a supplement.

The standard use of formal specification or models eliminates the ambiguity and thus reduces the possibility of errors introduced during software development. Of course, there is always the question of making a formal specification that matches to the customer’s actual consumption and it is difficult to state requirements tend to change during development.

We used formal languages for formal specification.

FSP is a Formal language based on pi-ADL. FSP is a process algebra test method for testing used concentrated description of the behavior of these components for the systems simultaneously. It provides the ability to formalize specifications software components and architecture. Each cell consists of Processes and methods have a limited number of states, and consist of one or more action. Formal verification test whether a model is a formally specify and verify and developed system meets its intended purposes of development [Bakar and Selamat,2011].

A behavior of concurrent programs depends on thei interaction with other programs. They have a collection of processes. The Labeled Transition System (LTS) provides a semantic checking for these concurrent programs, and it is mostly used for model checking for automatic software analysis.

We implement our system by using JADE (Java Agent Development Framework). The JADE platform allows the coordination of multiple FIPA compliant agents and the use of the standard FIPAACL communication language in both SL and XML. JADE is a middle ware between agents and environment of a system. It provides FIPA specifications and a set of graphical tools that supports the debugging and deployment phases.

The purposed system consists of autonomous agents, which is the user guidance system applies to JADE (Java Agent Development Environment). Each element is a JADE agent. With many elements, it consists of one or more subcomponents; each agent can make changes to components and changes can be implemented easily to each side of each component in the system under test, especially for critical components. This multiagent system informs a user about their exact location, and guides him to find other locations (i.e. faculties, offices, hostels, banks, other buildings etc) side the Bagdad-ul-Jaded Campus through Cartesian coordinates. By using agents user can find out their location along its X-axis or y- axis.

2. Objectives

The process of defining objectives of any system is important so that any management and user can understand the developed system and know how to tackle the developed system in order to fulfill their requirements.

2.1 Travel Guidance

We draw our system, which should provide a simple and transparent user interaction. Additionally, users should be able to use services with minimum interruption even when changing devices. In recent years, the technology factor has proven to be a reliable solution for many fields, including solutions such as application depends on its position related to tourism, for example, a multi-agent intelligent recommendation focuses on the dynamics changes according to of user location and user profile collected user preference. Using the list based on multi-agent system is a system that allows users to search for information and tourist activities such as data plan

2.2 Multi-agent (Distributed) Advantage as Compared to Centralized System

Multi-agent system provides a distributed system which works on Computer. A distributed system, in which a collection of autonomous computers are linked through a network. Each system of a user is appearing as a single system/computer. The Advantages of Distributed Systems over Centralized ones is its Incremental growth: Small increments can be added in computing power. It provide reliability if one machine crashes, rest of the system still working. Open system is the most characteristic point of a distributed system. It always communicates with other system as it is an open system.

2.3 Formal Methods

Our third objective is to provide formal methods for our developing system that give us control to reduce the complexity of the system by offering abstraction facilities, and they make unambiguous. Correct, precise and consistent specifications possible at lower cost. Formal methods are implemented on different aspects or characteristics of large system.

They are commonly used in our system to obtain a detailed description of the construction and testing of critical parts of large and small systems, safetycritical systems.

2.4 Formal Architecture – ADL

The fourth objective is to provide formal architecture for this user travel guidance management system. Formal architecture through the use of architecture description language (ADL) expressed their, semantics and grammar of mathematics. Formal architecture has an important role in the management of the security of the system by reducing the number of possible errors significantly.

2.5 Formal Verification

Our next goal is to formally verify our system to check with the formal verification of whether the system meets the requirements. We are in a formal verification of models, which are defined in a hierarchical structure and are consistent with how the human developer can engage. To officially confirm the design, you must first be converted to a simple form of validation. We fully comply with the requirements of our validation through a formal, easily formal methods are less restrictive and more flexible. Safety and liveness properties of the currently defined specifications at this time tested.

2.6 Formal Specification of Architecture

We are proposing a formal specification of system’s architecture, as determined, static and dynamic aspects of the system. All these architectural elements are identified separately and then combined to represent a linked system as a whole. We will reconsider the use of standard specifications source or target specifications described in such a way that it is possible that, generally, the specification of the source or destination. Formal specification language is a language with a formal definition of the syntax and semantics used as symbols or target specifications.

2.7 Proposing an Approach

Our goal is to offer a development based on a combination of methods and techniques for evaluation of formal specifications. An Approach that supports the analysis of functional properties and non-functional properties step by step to improve the abstract and concrete specification and formal verification of these specifications.

We analyzed the development process of our system by using a number of agent and focused on main stages of the specification requirements, testing requirements, architectural features and implementation

One of the most difficult tasks in software development for multi-agent systems is to ensure that the safety and liveness. The formal specification and verification of these systems plays an important role to ensure the safety and liveness of the system.

2.8 Implementation by Multi-agent System (JADE)

The system consists of autonomous agents who manage the user Location guidance System that is implemented by using JADE (Java Agent Development Environment). Each element is a JADE agent. With many elements, it consist of one or more sub-components, each agent is able to make changes in the component and these changes easily be implemented. So we have a system in which autonomous agents. We purposed a solution which with the identification of Components and subcomponents of the system, i.e., each part of the system can be formally defined.

3. Related Work

Agents perform such thing that can actually act so that why they are called agents [Shardlow, 1990].

Agents are simply computer systems that are in a particular environment and are able to act independently in this environment so they can achieve the goals of their design. An Agent usually sense its environment (Sensors are physical in case of active substances in the real world, or sensors content for software agents) and contains a repertoire of actions for non-deterministically response to modifying the environments [Wooldridge and Jennings, 1995].

Agent is a computer system that is capable of acting independent (autonomous) for its User or owners and objective of these agents is to achieve desired goals of designed system which is told by its owner/user. [Wooldridge, 2002].

A multi-agent system is a collection of several interacting agents in which each agent has in complete information or capabilities for solving the problem [Jennings et al., 1998].

Multi-agent systems provide access to resolve a problem of a software system by decomposing the system in to set of autonomous entities that incorporated with its environment to meet the quality system requirements and functionality **[**Weyns et al., 2006].

Multi-agent systems (MAS) are a family of systems. The control is not distributed in multi-agent system, i.e none of system components has control or global knowledge of the system. The components of demand multi-agent system are the agents. Agents have a Reflective nature it means that software components are capable of acting independently to fulfill their design. Decentralization of control means that agents must cooperate and coordinate their behavior in order to provide all the functionality of the system. [Weyns et al. 2009].

The Agent Modeling Language (AMOLA) used for modeling multi-agent systems and on the other hand the Agent Systems Engineering Methodology (ASEME) for developing multi-agent systems. AMOLA give us syntax and semantics to create models for multi-agent systems. It also provides help to covered design and analysis process for software development. It gives supports to modular agent design approach and provides new concepts of intra-and inter-agent control. It coordinates with different modules to define the behavior of agents that depicts agent's capabilities; latter defines the protocols that show the coordination of different agents [Nikolaos, 2009].

[Weyns et al., 2006] mentioned in their article,another perspective on software engineering with MAS. They provide an overview of the architecture reference best for MAS. These overviews of architecture integrate and extract some common function and structures of various learned and built applications. This reference provides architecture that provides application for MAS to share some base of the system. Given the nature of software architecture covered the way for mass integration with technology integration software.

Formal methods for development have received much attention in research centers, but for developing a large software systems they are rarely used in industry. One reason for this is that very little knowledge about the formal method's integration for software process and formal method's role for software life cycle is still unclear. In this work, a detailed examination of application is made, benefits and general formal methods (VDM) in a standard model for software development [Plat. et, al. 1992].

[Easter et al, 1998] described three case studies in their paper, they provide an application for requirements spacecraft fault protection systems modeling by using light weight formal methods. In many cases it is reported that formal methods have been applied to check the validity of requirement engineering process. To improve the formal specification, results is fed. We described in each case how formal methods applied, how much effort and what were the results of these methods. In all three cases, formal verification and validation can be improved by studying the characteristics of changes occur in methods. In the result we know that the early formation of unstable requirements; there is a need of efforts to maintain the cost of multiple representations.

Formal methods specification. These methods are used to describe the source or target specification, so that you can formally reason about the source or destination list. A formal specification i.e language that formally defined syntax and semantics, these notations used to formally define a source or target specification [Plat.et, al. 1992].

Description of the software architecture provides a formal specification of the architecture in terms of components and connectors and how they behave and composed together. In addition, a dynamic software architecture must specify that how to change the architecture of the software system at runtime. Allow specification of dynamic architectures is a big challenge for architecture description language (ADL) [Oquendo, 2008].

Formal verification test whether a model is a formally specify and verify and developed system meets its intended purposes of development [Bakar and Selamat, 2011].

Formal methods used for inspection activities in a formal manner. As a reference to the source and target specification, this target specification verified that setting is a correct translation of the source specification. A formal verification method provides conclusive evidence and proof system whose evidence may be excluded. The proof obligations make explicit what should be tested, and the proof system provides a mathematical framework for the construction of a proof is an important activity [Plat et, al.1992].

An architecture description, from a runtime perspective, should provide a formal specification of the architecture in terms of components and connectors and how they are composed together [Oquend, 2004].

Flavio Oquendo describes in his article π-ADL, a novel ADL that has been designed in the Arch Ware European Project to address specification of dynamic and mobile architectures. It is a formal, well-founded theoretically language based on the higher-order typed p - calculus. [Oquend, 2004].

π- ADL provides a graphical notation defined as a UML Profile in a model driven architecture framework [Brown, 2004].

Formal methods for software development received very attention to the academic environment. This mathematical Standard methods and specification enable the development of software systems and tools for the future development. They controlled complexity by offering an abstraction, and make clear possible and consistent data at low cost. [Plat et, al.1992].

First part of the design is to place intelligent agents in environment, through which they can sense their world. There are many ways to handle changes but more concentration is required for modeling any system. Agents are responsible for sensing and provide many advantages by making sense of the environment. Representation of perception allows many questions to be asked provide many software engineering advantages with respect to the system design and specification that is compatible with many accepted definitions of the provide [Heinze. C, et. al, 2001].

Formal specification is usually beneficial formal language specification; it is more concise and clear. These techniques will help us gain more knowledge about the system design, ambiguities, to maintain levels of abstraction, and also determine our approach to the problem and its implementation [Bowen and Hinchey, 1995]

Performance of the model testing of systems is sensitive within large system requires more time and memory for analysis. Moreover, parallel systems, the situation grows exponentially with multi-component and this is a state explosion problem. To check the model examples of industrial scale we will reduce the problem of situations room explosion. As a result, researchers have proposed model checking techniques for reducing the state space for use in many cases [Tip, 1995].

Dinkloh and Nimis describe a graphical tool that bridges the gap between planning and execution of JADE agents. It is implemented as a plug-in for Eclipse IDE and calls automatically translating AUML into a source code of jade represented using cpXML. It will also work to develop a methodology for the analysis and design of MAS with JADE [Dinkloh and Nimis, 2003].

Jade is a medium that simplifies the development of applications. Many companies already use it for various applications, including supply chain management, manufacturing, holonic, extraction, fleet management, auctions, tourism, and a number of articles in this special issue of the EXP journal already give evidence on the type of use, while this section attempts to describe the what works best in this section of the manual JAD. [Bellifemine.et al, 2003].

This white paper provides an overview of the JADE platform presents the architecture and main features, and describes a conceptual model based on jade. Two important aspects of the conceptual model shows: System distributed topology with peer-to-peer networking and software components with agent architecture. That defines how the various elements is to wait on the other side importance of standards for software interoperability, particularly, compliance also stressed. This paper also tries critical issue of technology transfer to address critical importance to such a transfer, when the search software and advanced business applications. The recognition that much of human technology transfer process map is aligned with the center of the organization and development of populations of Jade kernel extensions, and applications using jade infrastructure. Two major institutions of jade is the open source community and the Council JADE said. Finally, the main benefits of the Jade approach to take with the intention to help readers if JADE can meet the needs and areas between JADE applications can be very useful. [Bellifemine.et al, 2003].

Concurrent programs include a series of programs whose behavior depends on interaction of programs and responses to others environmental stimuli. Labeled Transition System (LTS) provides a model for semantic rules to capture the semantic operations of parallel programs, and is mostly used as a key point for software automated analysis such as tests based model and model checking (Zhenchang Xing, et al, 2011).

4. Background

In recent years, the agent technology factor has proven to be a reliable solution for many areas including the solutions depends on the situation, such as tourism applications. The important factor to using multi agent system is that some areas require it for system design. Especially when people or an organization with different goals and different information essentially wants to manage interactions of the agents.

Guidelines and procedures for the identification, technical analysis of expression notation - can be described as a formal method approach for developing a computer system that have a label with a well-defined semantics and syntax.

A formal method is an approach that describes the development of computer systems that includes a formal notation based on well-defined syntax and semantics, some guide-lines provided for using the notation, for analyzing the specifications of system that can expressed in the notation.

Formal methods are based on mathematics techniques to describe the system properties. It provides frameworks in which people can easily design and verify their systems in a systematic way rather than ad hoc. The word ‘formal’ is used in the sense that such a method is used a solid foundation of mathematics.

Formal verification helps confirm that your software models and code behave correctly. These verification methods rely on mathematically rigorous procedures to search through the possible execution paths of your model or code base to identify errors in your design. We are using Formal model-checking approaches for the architecture and verification specifications.

Model checking is a formal verification method. It gives a step by step process of a comprehensive survey of the mathematical model.This usually involves testing all states and transitions in the model,using smart techniques to the specific areas to consider whole groups of countries in a single operation and reduce the computation time. Primary advantage of the model checking is that it is often fully automated and disadvantage is that it is not general for large-scale systems.

Model approach provides the specification language facilities that define the user to specify a model system and its behavior in terms of mathematical structures, such as shelters, sequences, Cartesian products, maps and functions. Such methods can be used to describe sequential systems.

We used Model checking technique for analyzing an abstract representation of a system to determine the effect of one or more properties. More specifically, it was defined as a technique for finite systems state control algorithms simultaneously. Control model is used to increase the quality of verification and validation defines and control assets that meet all application requirements. We used light-weight formal methods to describe our system formally as they are easy and flexible to apply. Light-weight formal methods offer the advantages of formalism and simultaneously reduce the disadvantages of over-formalism. They are the implementation of formal methods in a degree-level depending upon need.

5. Our Case Study: User Location Guidance System

What is Our System?

We are going to develop a user location guidance system based on multi agents. This multi-agent system informs a user about their exact location, and guides him to find other locations (i.e. faculties, offices, hostels, banks, other buildings etc) inside the Bagdad-ul-Jaded Campus.

Our system is based on Cartesian coordinates. The coordinate system consists of axes oriented perpendicularly to each other, known as Cartesian coordinates. In addition to locating the position of user it also locates and guides him towards the other locations. We have exemplified our approach by taking a case study of the Baghdad-ul-Jadeed campus of The Islamia University of Bahawalpur as our area of implementation. Using the same approach we can very easily extend our case study to be used for the whole city or province or country or continent. User can easily interact with system through GUI.

To get access to the information from system users have to register themselves. After being registered, user will receive their User ID by user agents for future use. All information provided for registration will be recorded in database by database agent. Users enter their current position and find the other nearest position on their x-coordinates and Ycoordinates.

All information including User Information, Current Location and New Location with their coordinates along X-axis and Y-axis, user profile and Reports will be managed by agents. In addition it locates the position of user; it also locates and guides him towards the other locations.

At the end system will provide Reports that will be generated in graphical shape, reports. Number of User added, etc. All the data of this information will be stored in the database of the system and can be access at any time in need.

What is an Agent?

Agent is simply a piece of code, a person or simply a machine that can interact with its environment to fulfill the requirements independently.

A system that is capable to handle (Understand which type of requirements are to be needed to meet the development goals, and as long as he told me) independent action is called agent.

The role of an agent is that what you’re supposed to perform in the any organization, these are concert with many factors in relation to the organization itself. Role of an agent in term of its specific task is that they perform task that is related to organization as a whole. Organizational models accurately describe all functions that include IT organization. It is based on the characteristics, their activity, responsibility, and in terms of protocols and interaction models. Expressions of vitality and security play an important role in validating.

Why We Use Multi-agent System?

A multi-agent system is a paradigm for the study and development of distributed system consisting of individual agents, where each agent is autonomous. A multi-agent system is used to solve problem that is difficult or impossible to solve by centralized or monolithic system.

We used multi-agent system to improve efficiency and quality, and reduce costs and save time by using the tools available to solve many problems. For complex systems, Software engineering approach offered by interacting agents for developing applications. For software development multi-agent systems provides a large number of new abstractions and development compared to traditional software development approach. Therefore we required unique methods of analysis and design and new tools to effectively build our systems.

Why we used Formal verifications?

Formal methods are used for verifying and specifying the system’s properties. It has well defined syntax and semantics based on mathematics. Formal verification gives us surety that the developed system behaves correctly or not. These verification methods rely on mathematically rigorous procedures to search through the possible execution paths of your model or code base to identify errors in your design.

We are using Formal model-checking approaches for the architecture and verification specifications.

Model Checking is the primary strength in formal methods. In model checking technique, we were checking the desired property of the developed model. Model checking is fast, automatic, and supports partial specifications.

For Formal verification we used Labeled transition systems (LTS). LTS is a tool for testing systems simultaneously. Verifies that the mechanical characteristics of the modern system meet the characteristics required by his behavior.

Why we use Formal specifications?

Specification is the process of describing a system and its desired properties [David Parnas]. We used formal languages for formal specification. For this purpose we used Finite state process (FSP). FSP is a Formal language based on pi-ADL. In FSP we are using a process algebra test method for testing, used concentrated description of the behavior of these components for our purposed systems simultaneously. It provides the ability to developed system to formalize specifications software components and architecture.

Formal Specification:

Specification is the process of describing a system and its desired properties. A specification is a statement of properties required of a product, or a set of products [David Parnas]. We used formal languages for formal specification. For this purpose we used FSP.

Finite State Process

FSP stands for Finite State Process. It is the notation used to specify the behavior of concurrent systems to LTSA. FSP specifications generate finite Labeled Transition Systems. (FSP) is used as the input language and produces a transition day (LTS). FSP language based on process algebra. It is a formal language that can be applied to a system at different levels, from the implementation of specific formal request formal lightness. LTS is a discrete system operation and provides a log of actions. There are parallel processes with synchronization between them by action sharing. The goal is to specify our system by using light-weight formal method FSP notation along with LTS to prove correctness properties.

JADE (Java Agent Development Framework):

We used JADE (Java Agent Development Framework) for the implementation of our system as JADE (Java Agent Development Framework) is a software framework that is implemented in Java to make development process of multi-agent application easy in compliance with the FIPA specifications. JADE agent platform tries to keep high the performance of a distributed agent system implemented with the Java language.

For good runtime efficiency, reuse of software, agent’s mobility and knowledge about different architecture it uses agents models and implementation done in JAVA.

A system consisting of agents that communicates by using Agent Communication Language (ACL). FIPA has defined an agent communication language for specification. The specification consists of a set of message types and the description which is the effects on the mental attitudes of the sender and receiver agents.

The back-end of the project, which will keep records, will be designed in MySQL Server. We create tables in database that is connected with the Graphical User Interface (GUI).The front-end, which will provide Graphical User Interface (GUI), will be developed in Eclipse with JGUI plug in which provided a complete suite of controls to create database security for the application.

System Architecture

System architecture is a conceptual model, which defines the structure, behavior, and other views of the system. Formal

The back-end of the project, which will keep records, will be designed in MySQL Server. We create tables in database that is connected with the Graphical User Interface (GUI).The front-end, which will provide Graphical User Interface (GUI), will be developed in Eclipse with JGUI plug in which provided a complete suite of controls to create database security for the application.

System Architecture

System architecture is a conceptual model, which defines the structure, behavior, and other views of the system. Formal description and representation of the system can be specifying by system's architecture. It gives support of thinking about the structure of the system, including system components, the external features visible of these elements, systems organized relationships

We purposed a formal architecture that specifying the static and dynamic aspects of a developed system. This architecture, identify all architectural elements. Architectural elements are identified separately and then associated with the representation of system as one unit.

This architecture will show how a new system will function, to create a model you start with the analysis of existing and reapply the same thought process you used to create it, also integrate the user requirements for the new system. This architecture will show how a user/customer will receive information related to his location according to their axis. He asks for finding the location and process of extracting the location will be done very carefully. The result after tracking will be checked and report will be generating by report agent. Database agent will manage all the processing of data related to the exact location and all information of user/customer will be stored in the customer profile and administration. Formal verification will be performed at each level of the purposed system by using FSP.

The architecture of our system consists of Tourist Agent, Location agent, Profile Agent, interface agent, Report agent and database agent. Each representative agent receives data and information system data base using its database representative. Each agent has a relationship with JADE. Detailed processes of all the factors listed below:

- **Tourist Agent**

All users are assigned their own individual agents, called Tourist Agents. To provide services, it collaborates with the Database Agent to ensure that the model of the environment currently being monitored by the Database Agent is up-to-date. Each time an update is posted to the Database Agent, the Tourist Agent updates its own cache with the appropriate data.

- **Interface Agent**

The Interface Agent is the user's initial contact point. It provides the facility for initial registration process of agents (Tourist agent) to users. Each user has its own Tourist Id for contacting with Database agent for fetching information regarding to its location.

- **Profile Agent**

The profile agent provides information that is relevant to the user's location, all information is filtered using a model of the user's preferences and profiles he Profile Agent handles all the responsibilities of what type of data should be send to the user on demand in a right way about their location and other places near to its coordinate.

- **Location Agent**

Locating User position is the primary function of the Location Agent. Location agent uses Cartesian coordinates to locate and guides user towards the other locations. The coordinate system consists of axes oriented perpendicularly to each other.

- **Report Agent:**

This Agent will create automated statistical reports related to number of users wants to locate their position, number of users receive the service etc. This agent generates report by interacting with the different agents.

- **Database Agent:**

Database contains large data files that become more important some times. The Database agent manages all the resource information (Hostel, Faculties, Mosque, etc).It contains the information of user's location along their X-axis (latitude) and Y-axis (Longitude).All agents (Location, Tourist, Profile, Interface) interact with Database agent that is connected with Database for information retrieval

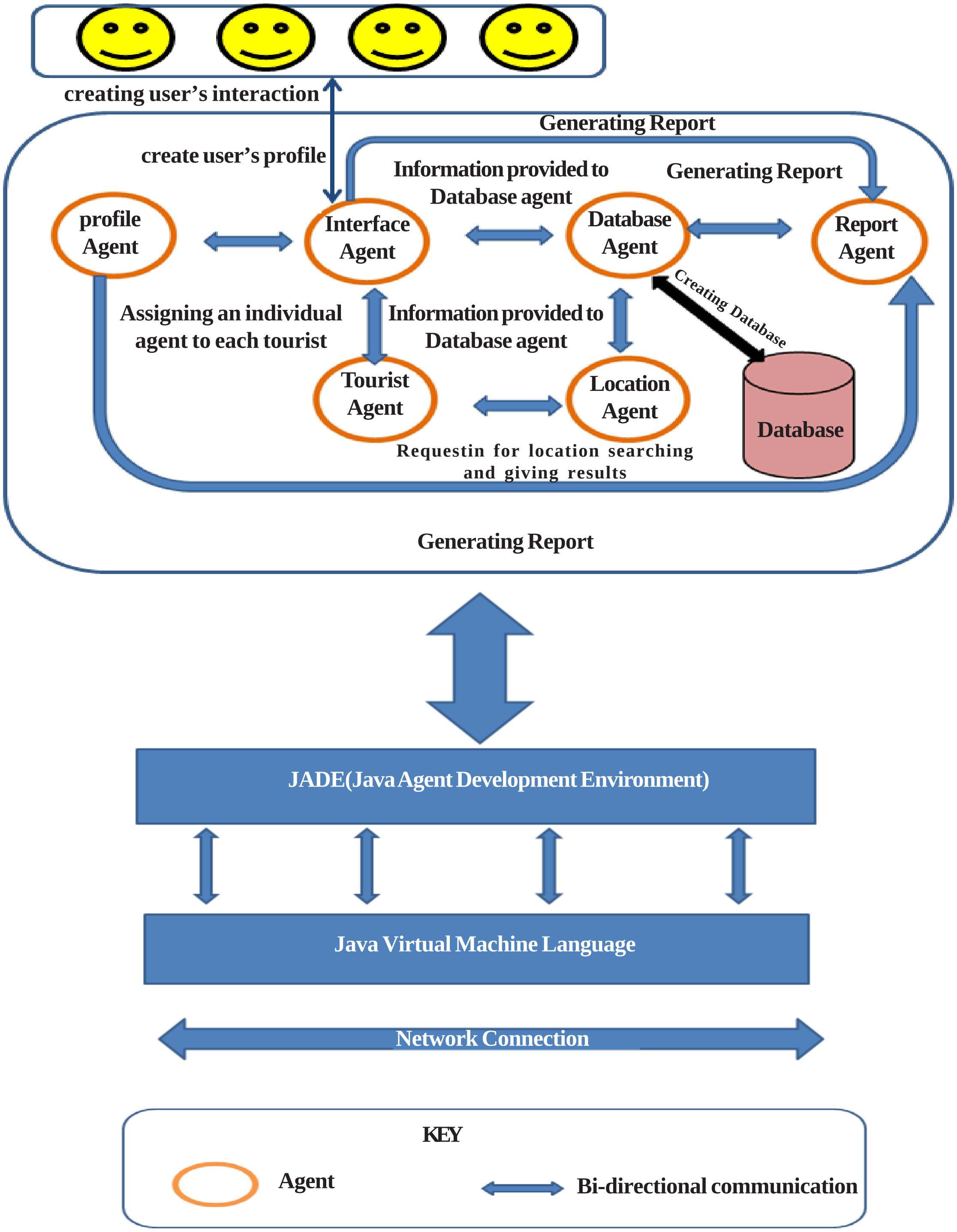
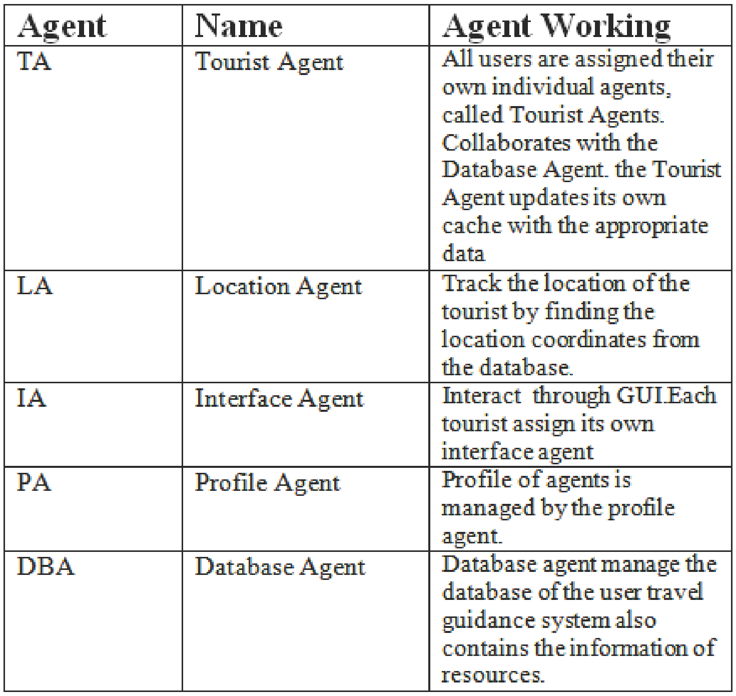

creating user's interaction
create user's profile
Generating Report
Information provided to Database agent
Generating Report
profile Agent
Interface Agent
Database Agent
Report Agent
Creating Database
Assigning an individual agent to each tourist
Information provided to Database agent
Tourist Agent
Location Agent
Database
Requestin for location searching and giving results
Generating Report
JADE(Java Agent Development Environment)
Java Virtual Machine Language
Network Connection
KEY
Agent
Bi-directional communication

FIgure 1. User Location Guidance System Multi-agent Architecture

Implementation:

The purposed system is based on Cartesian coordinates, A Cartesian coordinate system in a plane has two perpendicular lines (the *x*-axis and *y*axis) as shown in diagram. Our system finds nearest point of user's current location along longitude and latitude values. The following grid shows a point A (20, 15) as shown in the following diagram.



Agent	Name	Agent Working
TA	Tourist Agent	All users are assigned their own individual agents, called Tourist Agents. Collaborates with the Database Agent. the Tourist Agent updates its own cache with the appropriate data
LA	Location Agent	Track the location of the tourist by finding the location coordinates from the database.
IA	Interface Agent	Interact through GUI.Each tourist assign its own interface agent
PA	Profile Agent	Profile of agents is managed by the profile agent.
DBA	Database Agent	Database agent manage the database of the user travel guidance system also contains the information of resources.

Table 1. Agents Table with working

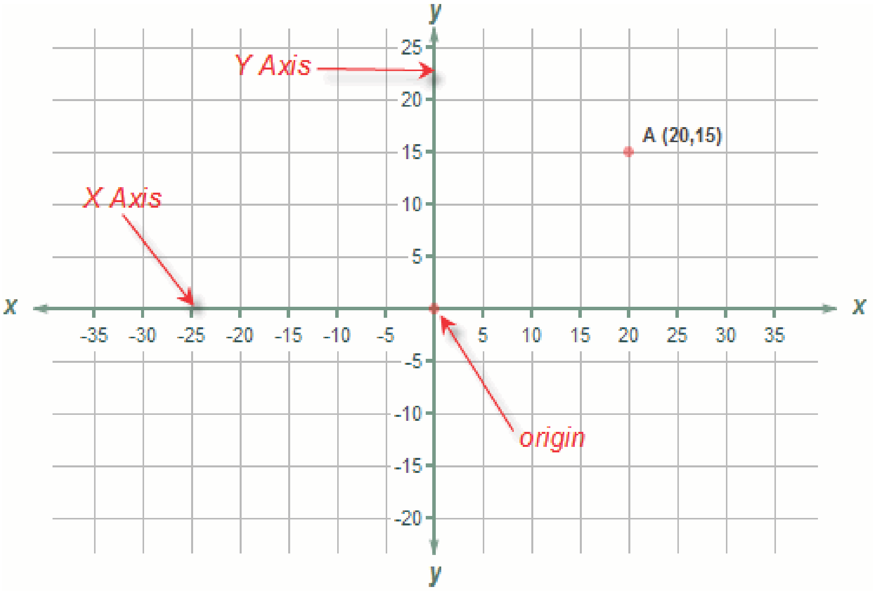
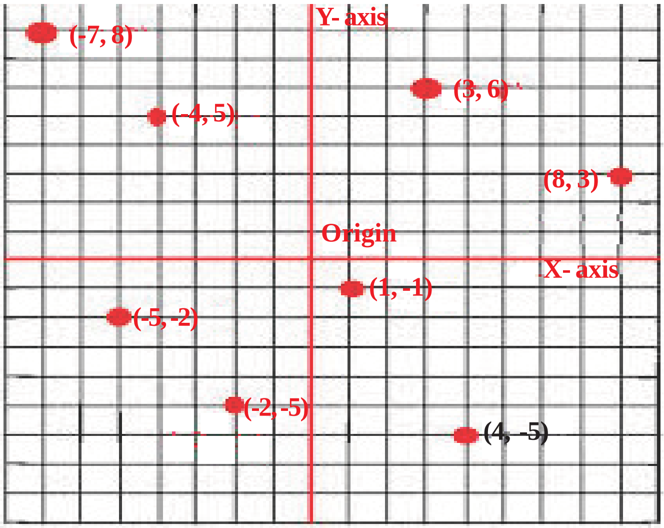


Figure 2. Working of the system

This diagram portray the working of our system, a point on a grid A (20, 15) shows nearest position 20 on longitude (x-axis) and 15 on latitude (y-axis) from user's current position (origin). It may provide more than two nearest point on a grid that can help user to travel in the Baghdad-ul-jadeed campus as shown in next diagram.

For implementing we used JADE as we know that JADE coordinates with many agents, each new user has an agent; each agent is able to make changes in the components and these changes easily be implemented. We propose a solution which begins with the identification of Components and subcomponents of the system, i.e., each part of the system can be formally defined.



The back-end of the project, which will keep records, will be designed in SQL Server. Longitude and latitude coordinates of system is saved in database. The front-end, which will provide Graphical User Interface (GUI) that, provided a complete suite of controls to create database security for the application. These include login control and the login status control that manages different view based on if the user is logged in (Hoffmann, 2003).

5. Conclusion and Future Work

In this paper, a multi-agent user guidance system is analysis, design, and then implementation. In this paper we used multi agent approach is based on multiple methods, models, and languages. It has modeling that is based on formal methods and lightweight formal methods implementations of formal methods.

In this approach we are using formal verification, architecture and then implementation of a formal system. These approaches also include formal verification by using fsp that checks its safety and liveness properties. Our purposed systems are a distributed system, multiple processes working in parallel and having synchronization between them. Methods and languages used jade, finite state process (fsp), labelled transition system (lts). in the future, there will be work done for the extension of this approach. The transformations from one to another model will be automated. Using the same approach we can very easily extend our case study to be used for the whole city or province or country or continent

References

[1] Bakar. N. A., Selamat. A. (2011). Analyzing Model Checking Approach for Multi Agent System Verification, Software Engineering (MySEC) ,5th Malaysian Conference in, p. 95 - 100, 13-14 December.

[2] Bellifemine, F., Caire, G. ., Poggi, A., Rimassa, G. (2003). JADE A White Paper. 3(3) - September http://exp.telecomitalialab.com.

[3] Bowen, J. P., Hinchey, M. G..(1995). Ten Commandments of Formal Methods. IEEE Xplore. 28 (4) 56 - 63.

[4] Brown A.W. (2004). An Introduction to Model Driven Architecture – Part I: MDA and Today's Systems. The Rational Edge, February.

[5] Dinkloh, M., Nimis, J. (2003). A Tool for Integrated Design and Implementation of Conversations in Multi agent Systems. *In*:Proceedings 2nd International Conference on Autonomous Agents & Multi agent Systems, Melbourne, Australia.

[6] Easter brook. S et al.(1998). Experiences Using Lightweight Formal Methods for Requirements Modeling, *IEEE Transactions on Software Engineering*, 24 (1).

[7] Hoffmann, H. J. (2003). NET: The Programmer's Perspective. *Journal of Object Technology*, 3 (2) 5-10.

[8] Jennings, N. R., Sycara, K., Wooldridge, M. (1998). A Roadmap of Agent Research and Development. Int. *Journal of Autonomous Agents and Multi-Agent Systems*, 1 (1) 7-38.

[9] Nikolaos, S. I. (2009). The Agent Systems Engineering Methodology (ASEME). Paris Descartes University, 2002, Paris, France.

[10] Oquend, F. (2004). -ADL: An Architecture Description Language based on the Higher Order Typed, *ACM-Software Engineering Notes* 29 (3) 1 to 14, May.

[11] Oquendo, F. (2008). Dynamic Software Architectures: Formally Modelling, Software Engineering Advances, 2008. ICSEA '08. The Third International Conference on, p. 352 - 359, 26-31 October.

[12] Plat, N., Katwij k, J. V., Toetenel, H. (1992). Application and benefits of formal methods in software,Software Engineering Journal, 7 (5) 335- 346, September.

[13] Shardlow, N. (1990). Action and agency in cognitive science. Master thesis, Department of Psychology, University of Manchester, Oxford Rd., Manchester M13 9PL, UK.

[14] Tip, F. (1995). A Survey of Program Slicing Techniques, J. Programming Languages, 3 (3) 121–189.

[15] Wooldridge, M. (2002). An Introduction to Multi-agent Systems by Michael Wooldridge. Published in February 2002 by John Wiley & Sons (Chiester, England).

[16] Wooldridge, M., Jennings, N. R. (1995). Intelligent agents: Theory and practice. Knowledge Engineering Review, 10(2):115-152.

[17] Weynsa, D., Helleboogh, A., Holvoet, T., Schumacher, M. (2009). The agent environment in multi-agent systems: A middleware perspective. *Multi-agent and Grid Systems – An International Journal* 5 (2009) 5, p. 93–108.

[18] Weyns, D., Holvoet, T., Schelfthout, K. (2006). Multi-agent Systems as Software Architecture another Perspective on Software Engineering with Multi-agent Systems, AAMAS, 06May 8–12, Hakodate, Hokkaido, Japan.

[19] Zhang, T., Covaci, S., Magedanz, T. (1998). OMG and FIPA Standardization for Agent Technology: Competition or Convergence, in Proceedings of the European ACTS Agent Workshop, Brussels, February.

[20] Heinze. C., Pearce, A., Sterling, L., Goss, S. (2001). Designing Perception Modules to Shape Information for Agents, Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader, 2112, p. 239-248.