JCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Itinerary and Mobile Code Patterns for Emerging Mobile Agent Systems in Large Scale Distributed Environments

R.S. Chowhan[1*], **P. Dayya**[2]

[1]SRF, Directorate of Extension Education, Agriculture University, Jodhpur, India
[2] Dept. of Ext. Edu. & Comm. Management, MPUAT, Udaipur, India

**Corresponding Author: word2rahul@gmail.com, Tel.: +9950463013*

Available online at: www.ijcseonline.org

Accepted: 22/May/2018, Published: 31/May/2018

Abstract— Patters in mobile agents help to design different applications which have non-permanent connections by adding the mobility code, improved networks, goal sustainability and machine based intelligence. Also the agent patterns mainly provide easy methods of developing the agent-based applications which are used or operated in open and large-scale distributed environments. It can provide various facilities like mobility, working in intermittent connections, autonomous execution, and many more that can allow sustainability of mobile agent based systems amidst of existing technologies like client-server. This paper gives an overview of the patterns of the mobile agents in mobile computing. Also the discussion covers the patterns and their usage, architectures, languages, applications, and the implementation challenges that are likely to be faced. This paper mainly discusses mobile agent which is a new technology that can be used in designing, implementing, and maintaining distributed systems. The discussion covers the patterns for mobile agents in the mobile computing applications.

Keywords— distributed computing, agents patterns, agent oriented programming

I. INTRODUCTION

Mobile agent is a program which moves through a network and are under their own control. The agents have the capability of moving from host to host where they have the ability of working with the other agents. The mobile, autonomous agents got the ability of providing a convenient programming pattern for different applications when they are partially connected computers which are involved. The mobile agent's helps in easily building dynamic and also distributed applications, the active objects are the ones which are referred to as agents because thus are autonomous and thus they are able to execute various tasks on behalf of the users. The computers which are partially connected are made up of laptops, digital assistants which are personified and the home computers, which are at point disconnected from a certain network. In this discussion, it covers the implementation and design agent system, Agent Tcl, and the features which provide support to the mobile computers and how the disconnected operational work [1]. The characteristics involve network-sensing tools and a docking system which enables an agent to be transparent.

Following the itinerary patterns of the agents it has led to an increased number of people who are using mobile connected devices like laptops, business computers and mobile phones. Mobile agents provides a robust programming, efficient, and convenient pattern for implementation of the distributed applications because of the features which they have such as mobility, autonomy, cooperation and intelligence. Pattern for Mobile agent is an effective paradigm in the area of distributed programming [2]. Mobile agents are faced with various challenges which, once they are met, and some internet sites accept mobile agents, the usage of the agents will spreads widely, thus leading to the revolutionizing of mobile computing.

The rest of the paper is organized as follows, Section I contains the introduction of mobile agent patterns and their various programming platforms, Section II contain the communication mechanism and network flow of agents, Section III contain the some advantages of agents, Section IV contain the application areas of agent technology, Section V explain patterns for mobile agent architectures methodology with flow chart, Section VI describes multiple agent system, Section VII contain the pattern of language support by mobile agent devices, Section VIII includes the mobile code patterns as applied in mobile agents, Section IX describe the mobile computing in mobile agents, Section X discuss mobile agent setup an its lifecycle methods, Section XI explain patterns of mobile agent's communication, Section XII explain structure and participating entities in mobile agent's patterns, Section XIII discuss challenges faced by itinerary patterns of mobile agents and Section XIV concludes research work along with future scope.

II. MOBILE AGENT NETWORKING AND FLOW

Mobile agent is a program which is autonomous and thus can come through the heterogeneous network when it's under its control thus it can migrate to different host and thus it has the capabilities of interacting successfully with different agents. Mobile agents decide where and when they should migrate, can execute at any given point, suspend the execution and move to another host where it continues with its execution. The features of mobile agents includes that it is autonomy, goal driven, intelligent, temporally continuous, learning, cooperation and also is creative. Due to the various features mobile agents are well adapted to the mobile computing domain. Due to the features that mobile agents possess they have the capability of moving from a PDA to internet and collect all the interested information that the user want. Mobile agents has also lead to simplifying the implementation, testing, and development of applications because they have they are capable of hedging the communication channels and thus they are able to show the computation logic. The agents distribute and then redistribute by a network and thus acts like a clients or also a servers based on the goals that they are serving [3]. They increase the scalability of application since they can move to the appropriate location as shown in figure 1 below.

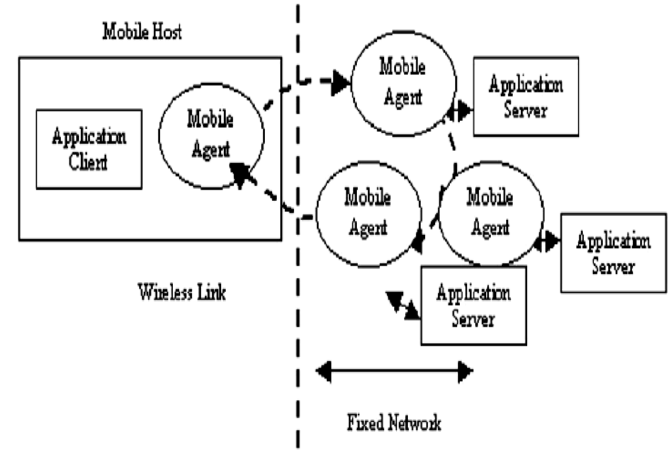

Figure 1. A flow diagram representing mobile agent's network

III. ADVANTEAGES OF EMERGING MOBILE AGENTS

Compared to the traditional server models mobile agents have many advantages. Mobile agents have led to the reduction of traffic in network. In most distributed systems performing a simple job may involve different multiple interactions which can result to increased network traffic. In the patterns of the mobile agent, the primary objective is taking the computation of data and this lead to consumption of fewer network resources and therefore there is increased efficiency. Mobile agents have also been able to overcome network latency this has been enabled because there is the management of critical time systems which have large sizes of networks that create latencies which are unacceptable in most cases [4]. Application areas of agent help in overcoming the problem because the agents can be executed locally upon the central controller's direction. Mobile agents also enable protocol encapsulation, the emerging communication protocols from the patterns of agents. Through this business can upgrade their protocols. Although the method is at time cumbersome, mobile agents help in resolving the problem because they migrate to the remote host and lead to the establishment of channels which are based on the proprietary protocols.

Mobile agents can perceive the various surrounding environment thus they have dynamic adoption which helps them in acting dynamically. Through the mobile agents, job which requires continuous open connections will result in cost savings. Mobile agents can have embedded tasks which can be removed in the network and thus can operate as independent agents. Also, there is seamless system integration since they have the aspects of both hardware and software thus the agents can give seamless system integration because they are dependent only in the environment where they execute [5]. Mobile agents can act and also react dynamically in the presence of even unfavourable conditions they make it easy to build a fault-tolerant distributed system.

IV. APPLICATION AREAS OF MOBILE AGENT TECHNOLOGY

Mobile technology has led to several applications which have brought many benefits. Agents have enabled electronic commerce, the agents can personify the creator's intentions, act, and also can negotiate on behalf of them thus making it to be well suited for the electronic commerce. Agents also acts as personal assistance to users and also can to per take tasks those using it on the remote host regardless of if a user is connected to a certain network. The technology for Mobile agent is a good solution for the brokenness in particular the context which is mainly for the collaborators who are untrustworthy. The parties that are interested parties can allow the agents to meet and are able to negotiate a secure host that is mutually agreed and form an alliance. Through the growing patterns and technology of agents there is distributed information retrieval in cases with a large amount of data instead of moving all the data to the search engine which creates search indexes [6].

The agents also provide an effective solution to managing advances in the telecommunication services through the provision of network reconfiguration of the advanced services in telecommunication. The agents provide dynamic network reconfiguration and also the user's customization. Because of the characteristics such as mobility, autonomous mobile agents provide autonomy to the workflow item and support the flow of information between co-workers. A mobile agent can also perform different tasks on the behalf of different users irrespective of whether a user is connected to a network or not. A user got the ability of dispatching the agent to the internet with the purpose of monitoring the prices of stock and they are notified once certain thresholds are reached. Mobile agents can also be applicable in the provision for administration of parallel processing tasks. In cases where computations require large amount of processing power, mobile agents help in getting the processes out. Mobile agents also lead to Information dissemination. The agents carry the upgrades and installation procedures directly to the user's personal computer [7]. This is achieved without any user's intervention i.e. it automatically updates and manages the software on the computer.

V. Patterns for Mobile Agent Architectures

Any task that is accomplished by an agent is in most cases configured in the first place. There are different configuration agents which includes; this is where a mobile agent does not have memory and thus it simply perceives the current situation get a rule which matches a certain situation and execute it. a reflex agent which have an internal state are the agent that perceives a current situation basing on the perception and what is found in the stored internal state which finds a rile which matches a certain situation and execute it. Another agent is the agent with explicit goals the agent chooses actions which can help it in achieving a goal. Utility based agent is the agent which maximizes some different performance measures while achieving it goals.

Simple reflex agent system

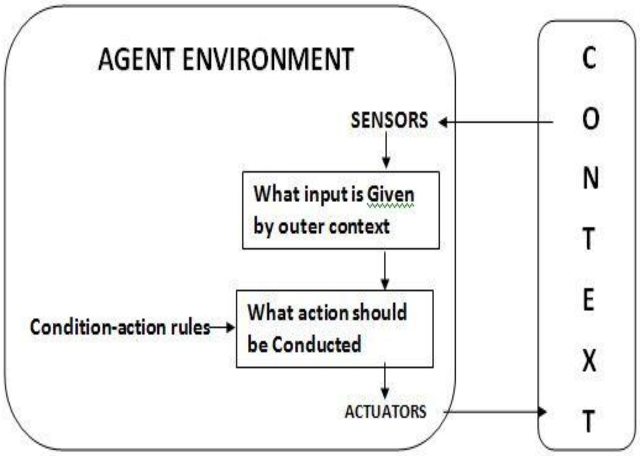

Figure 2. Agent Communication Environment System

The characteristic of the domain mainly determines the type of the agent that should be employed. These characteristics may include the number of agents which are needed, time, failure and communication cost, user's involvement, environment uncertainty, and also the probability of arriving at the set goal dynamically. A system can also have either a single or a multiple agent. A single system mainly employs use of one agent. The type of a system is more complex in comparison to the multiple agents at times when the job assigned is hard. In multiple systems, the control can be distributed and in the single agents does not have to complete the job given. In a single agent system an agent can be burdened unnecessarily if it is the one that has to complete all the tasks [8]. Single agent setting is mainly suitable for the system that has domains which requires a centralized control.

Multiple agents have different independent agents which can handle different separate tasks. Multiple agent system architecture supports the provision of the different parallel processing capabilities. In cases where one against fail, the system can still continue working because the agents shares responsibilities. The multi agent's systems also have the properties of capability where it is very easy to add a new agent to the multi-agent system as depicted in abov figure 2. Programmers can decompose a certain system into different tasks and then assign the tasks to various agents. Multi-agent system architecture mainly suits the systems which their criterion changes across different agents over time. Multi-agent systems can either be homogeneous or heterogeneous non-communicating multi-agent system.

VI. Multiple Agent's System

Homogeneous non-communicating multi-agent system is where all the agents have same goals, actions, knowledgebase and the same decision procedures. Heterogeneous systems, the agents which are employed have different goals, domain models, decision procedures and actions. In heterogeneous systems there is no communication the agent barely knows the goals, or even the decision procedures of another agents in the system. In heterogeneous communicating multi-agent system, the agent may differ in terms of goal, domain models, decision procedures, and actions as shown in figure 3. Communication among these agents mainly helps in coordination and effective resource allocation. Agents having the same goals but different abilities are thus organized into a team [9]. In that team each agent is thus assigned a role, the organization of the teams can be along the functional lines which are formed dynamically for the duration of a certain task as shown in figure 4.

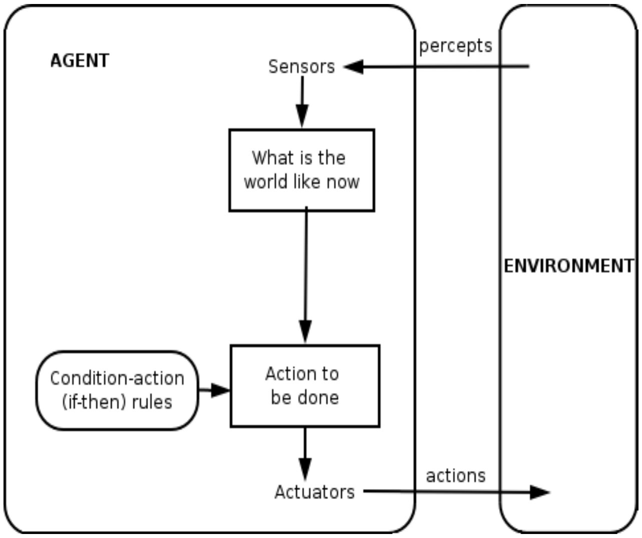

Figure 3. Perception of Multi-agent System

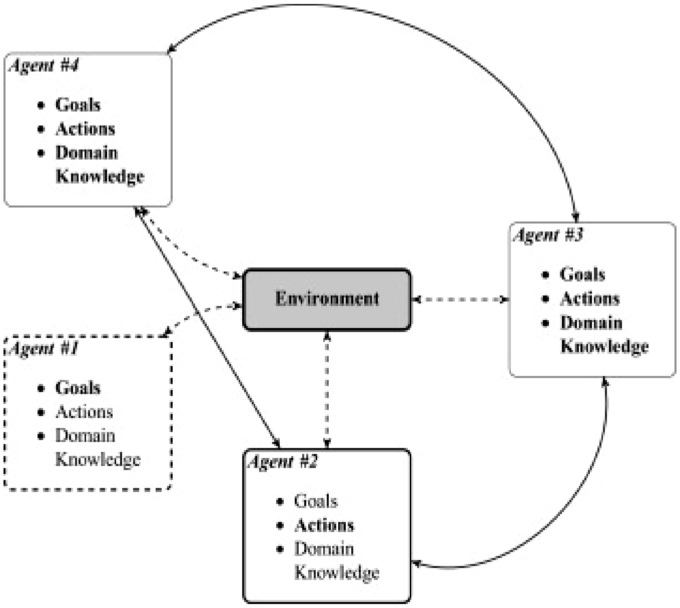

Figure 4. Homogeneous non communicating Multi-agent system

VII. Pattern of Language Support by MobilE Agent Devices

In implementing mobile agent number of new and open technologies like the distributed objects, java and the extensible mark-up languages are used. The most popular language used for implementing mobile agents is Java. In java, Voyager, Concordia, and Odyssey are implemented there. The features of Java like multi-platform support, object serialization, networking support which includes sockets, URL communication, distributed object protocol referred to as remote method invocation (RMI) makes it extremely suitable for mobile agent technology. Applets are used in launching and receiving mobile agents the XML documents are mainly used in publishing anything in the internet. It is one of the most popular accepted foundation layer on which to build. It can be used to encode data with meaningful structure and semantics that any agent with proper authorization can easily access, understand and interpret [10].

VIII. Mobile Code Patterns as Applied in mobile Agents

Java applets are the most used examples of mobile code. In the same case Java applets are not mobile agents this is because move once and upon the request of the user before they get executed.. Examples of true mobile agents include Typescript, Voyager IBM Aglets, Odyssey, Concordia, Agent Tcl, Tacoma, Mobile Service Agents and Ara. Mobile Service Agents are applicable in computing where they move to the user's machine.

A mobile agent is mainly transported by the means of Java Archive that have the serialized state of the agent. Before the agent is de-serialized and the linking of the classes is done in the servers the archive content mainly has to pass through the configurable pipeline of the security filters. The filter supports agent authentication and integrity checks. When an agent is administered to the server a group of thread is then created for this agent. A launcher is then gotten in the group of thread and it takes care of de-serializing the agent and it then is the first thread of the agent. In the pattern of data sharing, an agent requests for a migration through setting a ticket pointing to the desired destination. The server transport agents only when the threads in the agent groups have been terminated. The agent's beliefs are mainly based on the sensory input, when presented with an issue, an agent mainly reason to determine what they should do [11].

The patterns for mobile agents can be classified three classes that include travelling, task, and interaction. The classification makes it easier to understand the domain and application of every pattern help in distinguishing different patterns and also leads to the discovery of new patterns. Traveling patterns are the main essence for mobile agents these patterns include itinerary, forwarding, and ticket. The itinerary pattern is considered with the routing among different multiple destinations. The itinerary helps in maintaining a list of different destinations, defines the routing scheme, help in handling special cases. Another important pattern of travelling is the forwarding pattern it allows a certain host to forward all or main specific agents to other hosts. The ticket pattern involves an enriched version of the URL which embodies the requirements which concerns quality of services, permission, and other different data. It involves time-out information used for removing a certain agent to the remote host.

The task patterns are mainly concerned with the breakdown of different tasks and how the agent gives the mobility that is needed for performing tasks at different destinations. The plan gives reusability of different tasks, dynamic assignment of those tasks to the agents, and the composition of tasks. The interaction pattern helps the agents to communicate with each other the patterns are mainly concerned with locating agents and facilitating their interactions. The interaction patterns include Meeting patterns which makes a way more agents to initiate local interaction with a certain given host. The Locker defines a private storage space for data left by an agent before it is temporarily dispatched to another destination. The Messenger defines a surrogate agent to carry a remote message from one agent to another. The Facilitator defines an agent that provides services for naming and locating agents with specific capabilities. For instance, an information gathering agent continuously moves in the network and other different agents may wish to retrieve updates from the agents of information gathering without knowing the present situation. Lastly the organized Group Composes multiple agents into groups in which all members of a group travel together. This pattern is considered to be as a fundamental element of collaboration among different multiple mobile agents [12].

IX. Mobile Computing in Mobile Agents

Mobile agents are excellent paradigm which is used for implementing distributed applications, mostly in the context of connected computers. An agent is considered effective when it supports disconnected operations in different ways. The agents should be able to jump off to partially connected computers and also be able to return to it later. The agent also should be able to navigate through the internet. An agent must be able to sense and react to the network environment, so that it may act autonomously while its user is disconnected and also it must be able to communicate effectively to other different agents. Compared to the traditional client-server, the agents continue operating even when a laptop or a mobile agent is disconnected. The world of an agent tends to be uncertain and dynamic. The machines usually go up and down and there is a possibility that the information stored in the repositories may change and thus the exact sequence of destinations and the steps which are needed for completing an information gathering task might not be well known at the time when the agent is launched in the world. The agent is crippled without having an external state since it does not have a way of perceiving and adapting to the dynamic changes of the environment.

There are various sensors which allow an agent in determining the external state and different mechanisms which use these sensors for the adaptive navigation. Network sensing is the ability of a mobile to be able to detect the state of connection of a network. It performs an important task providing agents with information about the transit time across the different network whether a site is reachable or not. The information makes an agent to be able to adapt to the changing network conditions [13]. A smart agent is able to adapt to the fact that some sites are unreachable and thus can make the decision of having to visit the other sites first.

There are different network sensing tools which are used by the agent in gathering information about the status of a network. These tools include a way for determining if a local host is connected physically by using ping for broadcasting the address to the local subnet and if the response is within a very short interval then the network is considered as connected. Tools which determine if various hosts are reachable and the tools for determining the expected bandwidth to the remote host where the agents are able to choose their destination and the data amount basing on the bandwidth. Rather than measuring the bandwidth by sending lots of data to the remote host, which would often take as much time as sending the agent itself, we attempt to predict bandwidth from experience. The traffic monitors agent in every site is responsible for tracking information concerning all the recent communications that is given by the local agent server. The application agent contact the network monitor for obtaining the estimations of the bandwidth which are usually computed information which was recorded. The traffic monitors uses the weighted average of all the communications with the requested remote siteswhich weighted recent communications as being more heavily comparing to the older communications. If there are no recent communications with the requested site, the track monitor may use data from recent communications with similar" sites, that is, other sites in the same subnet or domain as the requested site [14].

X. Mobile Agent Setup and Lifecycle

Mobile agents are mainly transported using the means of Java Archives which mainly contains serialized state of the agent among different data. Before an agent is deserialized and a class is linked into the server the archive content pass through a configurable pipeline of the security filters. These filters mainly give support for agent authentication and integrity checks by digital signature means. When an agent is admitted to the server, a thread group and class loader is created for this agent. A launcher thread is then spawned in this thread group, which takes care of de-serializing the agent and later on becomes the first thread of the agent. An agent mainly requests for migration through setting up a ticket which is pointing to that desired destination and the server mainly transports the agents after threads in the agent thread groups are terminated. When the agent instance is de-serialized, its class name is compared to a property that is signed along with the agent's static part by the agent's owner as depicted in figure 5.

After de-serializing of an agent it is passed to all the lifecycle factories which are registered until the factory gives a signal that it is willing to use or handle the class of the agent. The factory generates a lifecycle instance which handles the agent's lifecycle, and wraps around the agent instance. The subclasses include the mobility context which provides methods for setting destinations tickets, access specific data for an agent, and also retrieves the agent's name. Communication context gives methods used for sending and receiving different messages. The environment is also a facility in the lifecycle which gives the dictionary directions on the space of an object which is shared with hierarchical name space for the keys. All the operations are subject to access control and the published objects are wrapped into proxy object which implement the varying degrees of separation between the called and caller objects [16]. The variable contexts give the read access to the agent's properties which are taken from the agent archive.

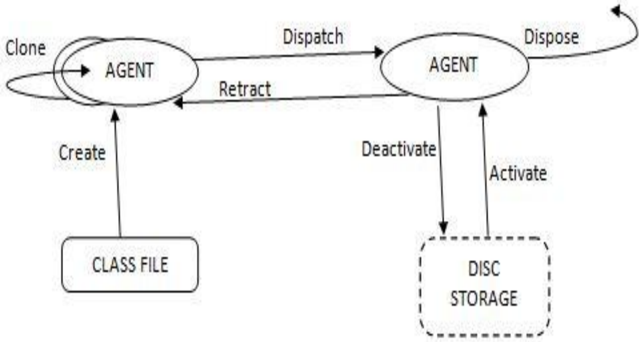

Figure 5. Mobile Agent life cycle methods

The navigation agents are applied in locating other agents which can serve the needs, access to the different dynamic index and also their locations. A uniform system of virtual yellow pages is used in helping the agent in deciding where they should go. These yellow pages have listings of services and their locations. By consulting these navigation services and using their network sensing tools, agents can formulate adaptive navigation plans to visit some of the services. Various adaptive learning methods are used in keeping the yellow update changes up to date. These include the application of new services register with one or more navigation agents in advertising their location. An application agent is also used in locating a list of navigation agent through querying the specialist agents onto the local host [17]. There are also agents which discover services accidentally and report all the corresponding sites to the navigation agent.

XI. Pattern of Mobile Agent's Communication

The agent Tcl now provides two major levels for the agent communication. The low-level mechanism mainly allows agents to communicate through passing message and also through direct connection which is established at the time when an agent issue meets the command and at the same time the receiving agent accepts the meeting. The higher-level agent remote gives call mechanism builds on top of the primitives thus adding structure and the higher-level abstraction to the communication. The server agents in the system registry have the local name server agent through specifying their interface in the flexible definition language. The clients' agent searches for a certain service through providing the same interfere and have the name server by finding a match from among the agent to the registered servers [18]. This flexible interface matching technique helps agents to exchange even when they share only a subset of a complex interface.

If a user is in need of using or want to have a certain task and thus requires gathering information through a mobile agent they have to submit the agent which is designed for performing that task to an agent submitter. The agent's submitters are capable for checking out the connections. If connection is ok with the specified Agent Host, where the user holds an account; then the Agent Submitter submits the Agent to it and gets an acknowledgment and can then go offline. In an Agent information is gathered from different host using the four agents which include the Pattern Database Generator's Agent, the pattern database processor agent, pattern based search and pattern based processor Agent. Pattern Database Generator Agent and Pattern Based Search Agent are the MAs whereas Pattern Database Processor Agent and Pattern Based Processor Agents is stationary agent. These agents maintains static as well as dynamic itinerary. In the static itinerary all parameter defined before the launching the agent this means that an agent know everything about the host while in the dynamic itinerary agent is able to know the information about the host; that is whenever required agent can be updated at any node at any time in the itinerary [19].

In the Pattern Based Search Agent if a user or client gave an input to a certain agent once they get the input agent starts its journey. The PBSA searches all the PV in PD on a node in which that pattern exists in the itinerary. This agent puts results on a node into the RC where the PBSA is ready to migrate to next destination in the itinerary. When agent visits its final destination it submits Result Container to Agent Reply Manager of the current visiting host and finally ARM sends result to the client of the agent. In the Pattern Database Processor Agent mainly processes the Result Container prepared by the PDGA during the itinerary. It retrieves three vectors from the Result Container which include the document vectors¸ pattern vector and frequency vector for every visited node in the itinerary. Pattern Based Processor Agent is responsible for processing the Result Container prepared by the PBSA during the itinerary. It retrieves three vectors from the Result Container for every visited node in the itinerary. By processing these vectors it regenerates all the original files and stores at local client node. Through these four agents the patterns of communication among the mobile agents are facilitated [20].

XII. STRUCTURE AND PARTICIPANTS IN MOBILE AGENT'S PATTENRS

The major participants in the pattern include the client, agent-view, user, agent, concrete agents, security managers and the execution place. The client is the one who manipulates agents through the agent view reference. A client can be the other agents or objects such as the java applets. The agent view is the adaptation of the proxy and the remote proxy patterns. The pattern is suitable for supporting transparent and also secure access to the different types of objects. The main objective of the agent view is providing transparent access to the agents. The access is done indirectly by the proxies so as to protect them and hide transparently their current localization. Also, agent view avoids the need to creating and managing remote classes. The user is mainly identified by an identity which is unique and contains their name, public key, certificates and the organizations that they belong to [21]. The user may have different identifiers which depend on the context that they are allowing the execution of mobile agent as depicted in figure 6.

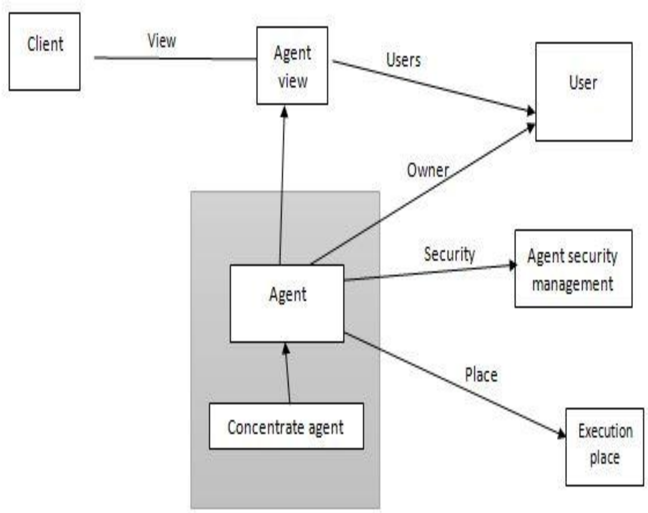

Figure 6. Participating Entities in Mobile Agent Structure

Owner of agent has the associated user identity which is represented by the user's instance. Various users can access the same agent by corresponding agent-view. The agent is the visible and extensible part of the agent pattern. The agent class has three main groups include the public final, call backs and the helper's methods. The final methods are operations which are provided by the agent's class which cannot be changed by a programmer. Call backs are the methods customized by specific agent classes and are invoked transparently. In this case events are triggered by actions which are started by the agent itself or by other entities that are related to it [22].

The concrete agent defines the helper's methods and the specialized call-backs which implement the specific functionality agents. The security managers control all the operations which are made available on the component of the agent through each agent view from its execution place [23]. This class specifies the agent's computational environment, which corresponds to the place where it was created as well as where it is currently resident.

XIII. CHALLENGES FACED BY ITINERARY PATTERNS OF MOBILE AGENTS

Mobile agents are faced with different challenges such as security issues. Security is a major issue which is needed to be considered in the implementation of the agents. Mobile agent has the ability of roaming galactic network which helps it in executing a code in a foreign server system. This ability makes it vulnerable to the malicious attack from the other agents and serves [24]. Another major challenge faced is protection of hosts from the malicious agents. Mobile agents are open system which can be easily attacked. Attacks are in forms of leakage, stealing and tampering of resources of resources and vandalism. A host can execute mobile agent code thus the agent has the access to the resources of the host. The access gives a mobile agent the power of attacking other local agents, can also generate viruses and deny the services from the other agents. Some of the solutions provided include authentication, verification, authorization, digital signatures, and provision of hooks where an encryption system may be added, enforcement of access restrictions such as sandbox architecture, proof carrying code or some other restrictions like CPU or memory usage [25].

Another challenge faced is the protection of agents from the malicious host. It is very hard to protect an agent from the malicious host than protecting a host from the malicious agent. Hosts executes mobile agents and thus can see the agent's data and code this makes it more easily for the host to tamer the agents code or to terminate them. Ways of protecting the agents from the host include limiting the amount of confidential data supplied to the agent and enforcing good host behaviours. The code compiled into some sort of platform independent representation, such as java byte codes is either executed inside an interpreter or compiled into native code, when it arrives at the target machine. Performance and scalability is also another issue, for portability and security purposes, agents are written in a slow interpreted language [26]. With the surfacing of just-in-time compilation code such as java, software fault isolation etc. some of the problems with the slower execution of mobile agent code are alleviated. However, several other problems involving migration overhead and slow execution still remain to be addressed.

XIV. CONCLUSION

Mobile agent patterns are considered as to be the most effective and growing patterns in the area of distributed programming. The current trends have indicated the widespread usage of the mobile agents in near future although various technical and also non-technical challenges are still not solved. The implementation mainly entails the application of public infrastructures, symmetric and asymmetric keys, cryptographic hashes and, in general, an extensive knowledge of cryptographic application programming. In general, the growth of mobile agent itinerary patterns has contributed to the stability of the software architecture, mobile agents are easier to adapt to the new requirement regarding the changing environments in the field of technology. Platforms are relieved from having to deal with different protection protocols of different agents thus the platforms' code does not need to be updated whenever a protocol is improved or a new one has to be supported. Most applications using mobile agent technology require the use of security mechanisms which are complex to implement. It is significant to have the mobile agent software architecture which is based on providing agents with the code that manages their own protection and execution thus it may improve the security measures for keeping the mobile agents safe during any operations.

REFERENCES

[1] Ahila, S. Sobitha, and K. L. Shunmuganathan. *"Overview of mobile agent security issues—Solutions."* In Information Communication and Embedded Systems (ICICES), International Conference on, pp. 1-6. IEEE, 2014.

[2] Hasan, Ragib, Md Mahmud Hossain, and Rasib Khan. *"Aura: An iot based cloud infrastructure for localized mobile computation outsourcing."* In Mobile Cloud Computing, Services, and Engineering (MobileCloud), 3rd IEEE International Conference on, pp. 183-188. IEEE, 2015.

[3] Chowhan R. S., *"Mobile Agent based Workflow Management System (MAWFMS) for Information Flow and Business Processes."* Orient.J. Comp. Sci. and Technol;Vol.11(1), 2018

[4] Roman, Rodrigo, Javier Lopez, and Masahiro Mambo. *"Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges."* Future Generation Computer Systems 78 pp. 680-698, 2018.

[5] Kogan, Alexander, Ephraim F. Sudit, and Miklos A. Vasarhelyi. *"Continuous online auditing: A program of research." In Continuous Auditing: Theory and Application"*, pp. 125-148. Emerald Publishing Limited, 2018.

[6] Xu, Yingyue, and Hairong Qi. *"Mobile agent migration modeling and design for target tracking in wireless sensor networks."* Ad Hoc Networks Vol.6, no. 1 pp.1-16, 2008.

[7] Cai, Wei, Min Chen, Takahiro Hara, Lei Shu, and Taekyoung Kwon. *"A genetic algorithm approach to multi-agent itinerary planning in wireless sensor networks."* Mobile Networks and Applications Vol.16, no. 6, pp.782-793, 2011.

[8] Gavalas, Damianos, Ioannis E. Venetis, Charalampos Konstantopoulos, and Grammati Pantziou. *"Mobile agent itinerary planning for WSN data fusion: considering multiple sinks and heterogeneous networks"* International Journal of Communication Systems Vol.30, no.8, 2017.

[9] Wang, Xiaofei, Min Chen, Taekyoung Kwon, and Han-Chieh Chao. *"Multiple mobile agents' itinerary planning in wireless sensor networks: survey and evaluation.",* IET communications Vol.5, no.12 ,pp.1769-1776, 2011.

[10] Chen, Min, Wei Cai, Sergio Gonzalez, and Victor CM Leung. *"Balanced itinerary planning for multiple mobile agents in wireless sensor networks."* In International Conference on Ad Hoc Networks, pp. 416-428. Springer, Berlin, Heidelberg, 2010.

[11] Cai, Wei, Min Chen, Takahiro Hara, and Lei Shu. *"GA-MIP: genetic algorithm based multiple mobile agents itinerary planning in wireless sensor networks."* In Wireless Internet Conference (WICON), The 5th Annual ICST, pp.1-8. IEEE, 2010.

[12] Chen, Min, Laurence T. Yang, Taekyoung Kwon, Liang Zhou, and Minho Jo. "Itinerary planning for energy-efficient agent communications in wireless sensor networks." IEEE Transactions on Vehicular Technology Vol.60, no.7, pp.3290-3299, 2011.

[13] Aloui, Imene, Okba Kazar, Laid Kahloul, and Sylvie Servigne. "*A new Itinerary planning approach among multiple mobile agents in wireless sensor networks (WSN) to reduce energy consumption.*" International Journal of Communication Networks and Information Security Vol.7, no.2, 2015.

[14] Cai, Wei, Min Chen, Takahiro Hara, Lei Shu, and Taekyoung Kwon. *"A genetic algorithm approach to multi-agent itinerary planning in wireless sensor networks."* Mobile Networks and Applications Vol.16, no.6, pp.782-793, 2011.

[15] Nestinger, Stephen S., Bo Chen, and Harry H. Cheng. *"A mobile agent-based framework for flexible automation systems."* IEEE/Asme Transactions on Mechatronics Vol.15, no.6 pp.942-951, 2010.

[16] Miller, Naomi Liora, Harold Roy Miller, and Warren Stableford. "*Translation of user requests into itinerary solutions."* U.S. Patent 9,659,099, issued May 23, 2017.

[17] Rahul Singh Chowhan and Rajesh Purohit, *"Study of mobile agent server architectures for homogeneous and heterogeneous distributed systems."* International Journal of Computer Applications Vol.156, no. 4,pp.32-37, 2016.

[18] Rahul Singh Chowhan, Amit Mishra, and Ajay Mathur, *"Aglet and kerrighed as a tool for load balancing and scheduling in distributed environment."* In Recent Advances and Innovations in Engineering (ICRAIE), International Conference on, pp.1-6. IEEE, 2016.

[19] Bagga, Pallavi, and Rahul Hans. *"Applications of mobile agents in healthcare domain: a literature survey."* International Journal of Grid Distribution Computing Vol.8, no. 5, pp.55-72, 2015.

[20] Dinh, Hoang T., Chonho Lee, Dusit Niyato, and Ping Wang. *"A survey of mobile cloud computing: architecture, applications, and approaches."* Wireless communications and mobile computing Vol.13, no.18 pp.1587-1611, 2013.

[21] Jara, Antonio J., Miguel A. Zamora-Izquierdo, and Antonio F. Skarmeta. "Interconnection framework for mHealth and remote monitoring based on the internet of things." IEEE Journal on Selected Areas in Communications Vol.31, no. 9 pp.47-65, 2013.

[22] Shiraz, Muhammad, Abdullah Gani, Rashid Hafeez Khokhar, and Rajkumar Buyya. *"A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing."* IEEE Communications Surveys & Tutorials Vol.15, no. 3, pp.1294-1313, 2013.

[23] Baig, Mirza Mansoor, Hamid Gholamhosseini, and Martin J. Connolly. "A comprehensive survey of wearable and wireless

ECG monitoring systems for older adults." Medical & biological engineering & computing Vol.51, no. 5, pp. 485-495, 2013.

[24] Iwaya, Leonardo H., Marco AL Gomes, M. A. Simplício, T. C. M. B. Carvalho, Cristina K. Dominicini, Rony RM Sakuragui, Marina S. Rebelo, Marco Antonio Gutierrez, Mats Näslund, and P. Håkansson. *"Mobile health in emerging countries: a survey of research initiatives in Brazil."* International journal of medical informatics Vol.82, no 5, pp.283-298, 2013.

[25] Bhaskar, B., Kumar T. Jagadish., Kamal, M.V., *"A Security Determination-Reaction Architecture for Heterogeneous Distributed Network."*, IJSRCSE, Vol.5, Issue.5, pp.22-34, 2017.

[26] Singla, Annie., Jain, Kamal., Gairola, Ajay., *"Delving into Security of Networks-Time's Need"* IJSRNSC, Vol.2, Issue.3, pp.1-8, 2014.

Authors Profile

Mr. R S Chowhan pursed Bachelor of Technology from Jodhpur National Univeristy, Jodhpur in 2012, PG-Diploma CDAC from IACSD, Pune in year 2013 and Master of Engineering from M.B.M. Engineering College, Jai Narain Vyas University, Jodhpur in year 2016. He has worked with many MNC's like Webmatrix Tech, Pune and Devlopers Trinity, Jodhpur, He has also been taking guest lectures in Sardar Patel Univeristy of Police, Security and Crime Justice. He is currently working as Senior Research Fellow in Directorate of Extension Education since 2017. He is a life member of International Society for Development & Sustainability since 2017. He has published more than 20 research papers and articles in reputed international journals like Oriental Journal of Computer Science and Technology and has also participated in various International Conferences conducted by IEEE and Springer, it's also available online. His main research work focuses on Mobile Agents, Network Security, Regression testing, Autonomous Load Balancing Mechanisms, Deep Learning, IoT, Parallel Comptuing, Agent-Oriented Distributed Programming and Computational Intelligent Agents based education. He also possess teaching and research experience in his field of interest.

Mrs P Dayya pursed Bachelor of Science and Master of Science from Maharana Pratap University of Agriculture and Technology, Udaipur, Rajasthan in year 2013 & 2015 respectively. She is currently pursuing Ph.D. at Maharana Pratap University of Agriculture and Technology, Udaipur, Rajasthan since 2016. She has published more than 15 research papers and articles in reputed international journals and it's also available online. Her main research work focuses on Mobile Agents, Entrepreneurial Activities, Adoption Income Generation, Agent Itinerary Simulation and Web Designing based education. She also has teaching and research Experience in her discipline.