

Topic Based Agent Migration Scheme via Publish/Subscribe Paradigm

Mustafa Akif Karzan and Nadia Erdogan

Abstract—Multi agent systems provide distributed computation for decentralized software applications. Mobile agents play an important role in realizing distributed systems. Current mobile agent implementations force a tightly coupled mechanism for migration by providing a mobile agent its destination address in advance. This requirement is both restricting and inflexible. In this paper, we propose a novel agent migration scheme that eliminates destination address information from the essential components of migration. Basically, an agent migrates between platforms as a specific type of a message over a communication system that implements topic based publish/subscribe paradigm. This approach allows for loosely coupled inter-platform mobility due to the communication model, resulting in a flexible and scalable execution environment. The paper also describes a new service oriented architecture for decentralized, peer-to-peer systems an information expert ecosystem, as an application of the proposed scheme.

Index Terms—Multi agent systems, publish/subscribe paradigm, agent migration, distributed systems, service oriented architecture.

I. INTRODUCTION

Developments in software technologies over time strongly affect architectures of systems. Centralized structures turn into decentralized structures, where software solutions are built as distributed systems rather than localized systems. Also, the distributed computation architecture is evolving from the client server model to the peer to peer model. This transformation is mainly due to scalability and performance requirements. Distributing computation from a single highly capable entity to multiple relatively less capable entities has proven to be much productive.

Programming paradigm has also evolving from a function centric model to an object centric one. Currently, we see the object oriented programming further evolve to give rise to agent oriented programming which enables the implementation of multi agent systems, a very popular research area.

Software agents are autonomous, collaborative and reactive software entities that act on their user's or owner's behalf. Mobility is an optional property of a software agent in multi agent systems; agent mobility is an important factor to

be considered because there are a lot of application models which require autonomous and intelligent computational power to run locally. Ametller, Robles and Borrel [1] indicate the importance of agent mobility especially in e-commerce applications. They point out the lack of mobility in agent systems or the restricted and non standard way of implementing mobility. They proposed to move agents around via FIPA (Foundation of Intelligent Physical Agents) ACL (Agent Communication Language) messages in a standard way. In [2] Malik, *et al.* proposed another way to achieve inter platform agent mobility. In their study they mention about converting one form of agent belong to one platform to another that is compatible with any other platform to migrate to. Because there is not a single standard way for agent migration, researchers continue to work on this field to achieve an effective agent migration.

All existing work for mobile agents tightly couples hosts that take place in migration also mobile agents and target hosts because existing migration process includes destination address to be given in advance. If the address is not provided migration cannot occur.

In this paper we propose a decoupled, topic based agent migration without informing host and mobile agent with the destination address. Also inter platform mobility is achieved in a standard way. What we propose here is another way of service oriented architecture with mobile agents in a multi agent system.

Achieving decoupled communication for mobile agents requires topic based migration without knowing the destination host address in advance. This mechanism is very familiar in publish/subscribe model. In this paper we get mobile agent concept with publish/subscribe paradigm converged to produce a scalable, peer-to-peer and decentralized system with decoupled communication basis. In literature there are studies [3]-[5] using both publish/subscribe model and mobile agents. They were interested in building publish/subscribe mechanics with middleware services. However in this study we are interested in publishing mobile agents themselves to the destination hosts with the help of a provider to achieve agent migration from one host to another via publish/subscribe model. So this is the first study to migrate agents between hosts and platforms using publish/subscribe paradigm which offers a new software architecture for decentralized, service oriented systems.

In this paper we propose an ecosystem called information expert agents ecosystem that includes information expert mobile agents, source and target hosts communicating over publish/subscribe model. In this ecosystem there are service producing hosts with mobile agents and service consuming

Manuscript received November 25, 2012; revised March 18, 2013.

M. A. Karzan is with the Department of Computer Science, Informatics Institute, Istanbul Technical University, Istanbul, Turkey (e-mail: karzan@itu.edu.tr).

N. Erdogan is with the Department of Computer Engineering, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey (e-mail: nerdogan@itu.edu.tr).

hosts. Service is provided via information expert mobile agents where those agents are published to topics of a messaging middleware. Consumer hosts subscribe to topics that they interest. Messaging middleware notifies subscriber hosts with mobile agents when producer hosts publish to the topic. With this approach a mobile agent migrates from source host to target host without knowing the destination address which identifies the network location of target hosts. Applying publish/subscribe paradigm system automatically becomes scalable. Multiple and dynamically changing destination hosts can be notified. This achieves decoupled inter platform agent migration and also a new service oriented architecture for decentralized, peer-to-peer systems via mobile agents.

Organization of the remainder of this paper is as follows. Section II talks about publish/subscribe paradigm and why we adopt it in our research. Section III is about mobile agent concept. Section IV explains system architecture how we design this communication infrastructure. Section V defines implementation considering used technologies and section VI gives conclusion.

II. PUBLISH/SUBSCRIBE MODEL

Publish/Subscribe is a communication/messaging pattern well adapted to deployment of scalable and loosely coupled systems. It integrates producer and consumer applications. It offers scalability and dynamic network topology support. Applications attached to the system do not know the overall network topology and even they are not notified with the information of the entities newly attached to the system or detached from the system. Publish/Subscribe paradigm involves messaging middleware, producers and consumers. In this communication system, consumers subscribe to events or a pattern of events that they are interested in. Producers publish the events that they generate and messaging middleware or message broker notifies interested subscribers of events when they are published by the producers via store and forward function. As stated in [6] Publish/Subscribe is an event based messaging where it varies by subscription schemes such as topic-based and content-based. Topic based scheme groups subscribers of same type of event of interest. Content based scheme improves topic based scheme subscription by considering the actual content of the event of interest.

This interaction paradigm provides decoupling in space, time and synchronization between producers and consumers of information. Space decoupling involves preventing producers and consumers interact directly. Instead they communicate over a message broker. Thus producers do not know how many subscribers are there that are interested in their events or even if there exists any subscribers. Also subscribers do not know who is publishing the event of interest in advance even if there is any. Decoupling in time provides both producers and consumers must not be active at the same time which means a producer can publish its events when subscribers are not connected or subscribers can be notified when producer is not connected. Synchronization decoupling prevents publish and notify actions to be in a synchronized manner. Thus producers are not blocked on

producing an event or subscribers are not blocked on notification they can get notified asynchronously.

Separating direct interaction of producers and consumers of information with a neutral mediator provides loose coupling in space, time and synchronization, increases scalability and offers flexibility in terms of dynamic network topology.

In this research we prefer publish/subscribe paradigm as a communication infrastructure to produce a loosely coupled, scalable and flexible system. With the adoption of this pattern, our system inherits these properties naturally. Also we propose an ecosystem with mobile agents that realize a topic based migration scheme. This allows several distinct topics and several publishers to same or different topics. Consumers may interest in multiple topics. This again fits best with publish/subscribe pattern which brings consumers and producers together by subject of interest not by physical network address. Consumers know what they need to consume but they may not know who provides it. Producers know what they produce but may not know who needs it. So, meeting point is a topic based intermediary to serve or get served without knowing (having reference to) each other in advance.

III. MOBILE AGENT CONCEPT

Mobile agent is an intelligent computation that can move within the network or in other words can migrate from one computer to another on behalf of the user or another entity. Mobile agent has autonomy, social ability, learning, reactivity and mobility properties. Mobile agent transports its code and data (state) to target host and continues its execution on its last state. Mobile agent concept differs from process migration systems by its autonomy and reactivity. Mobile agents act different on different hosts according to the state (data) of the host.

Mobile agent concept converts client/server paradigm to relocatable computation which reduces network traffic especially there are vast amounts of data to be processed. It provides distributed peer to peer systems rather than centralized systems. Asynchronous execution on different hosts is achieved. Target host may execute other tasks while migrated agent is executing its tasks. Moreover mobile agents adapt dynamically to the execution environment. They do not take same actions for distinct hosts if it is not required. They take actions according to the state of target host. Mobile agent concept also provides some level of fault tolerance on dynamic network topology depending on connections between hosts. Furthermore changing agent's source is enough to maintain this mobile computation. It does not require any configuration change on target hosts. This brings flexibility on update procedures. Real time processing capabilities are improved when executing locally rather than communicating with a remote host and executing afterwards and communicating results with remote host again [7].

Mobile agents and multi agent systems are well adopted when designing distributed peer to peer systems by their nature. In our research we prefer mobile agents because of these properties plus mobile autonomous computation needs. Especially when we consider e-commerce applications

depending on the vast amount of data to be processed, local computation is required due to efficiency concerns for many aspects like network load, real time operations ...etc. Mobile agents look best as a solution for this type of problems when their autonomy, reactivity and mobility features are noticed.

Publish/subscribe paradigm and mobile agent concept are very coherent when we look at their properties. Both suit well with distributed and peer to peer systems which increase productivity, scalability and flexibility to dynamic network topologies. In our research we propose a communication infrastructure bringing publish/subscribe paradigm and mobile agent concept together to produce a decoupled, scalable and flexible system with autonomous, reactive and mobile software entities.

IV. SYSTEM ARCHITECTURE

The system implemented allows for migration of agents between hosts by means of the publish/subscribe communication mechanism. In particular, agents are transmitted from a source host to a target host in the form of ACL messages. Due to the communication paradigm, both the source and the target hosts are completely decoupled. Because we perform topic based publish subscribe scheme, source and target hosts do not need to know each other in advance. They only know the mediator between them. Publish/Subscribe pattern makes this system decoupled in space, time and synchronization. Also system becomes highly scalable and flexible to dynamic network topology. Instead of sending transactions between hosts sending agent itself reduces the network traffic while increasing the system performance. Thus new type of mobile agent concept emerges. System architecture is shown in Fig. 1.

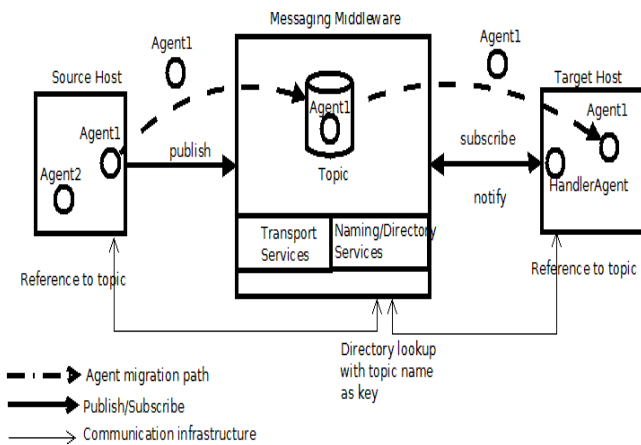


Fig. 1. System architecture overview.

A source host acts as a service producer and it is capable of generating agents that can present specific services. After a service providing agent is started, it publishes itself to a predefined topic on the messaging middleware. A target host is a service consumer that needs the help of service providing agent for its own purposes. A target host must subscribe to topic(s) on the messaging middleware to be notified of service providing agents. Subscription and notification handling operations are handled by a local agent, called HandlerAgent, on behalf of its host. A HandlerAgent is expected to be aware of its host's intentions; it subscribes to

the topics its host is interested in. When a service providing agent is published on a subscribed topic, the messaging middleware notifies the HandlerAgent with an ACL message that carries the service providing agent as a payload in a special form. The HandlerAgent extracts the service providing agent from ACL message and carries out operations that activate the agent on the local host. From this point on, the service providing agent evaluates the conditions and state of this new environment, provides a service specific for this local host and terminates when it is done. In this manner, the agent travels from one host to another in the form of a special message, without the need to know the identification of the target in advance. Source and target hosts are connected via a mediator through selected topic(s). In other words, a topic on the mediator is the meeting point for the service producing and service consuming hosts. Hosts realize directory lookup on the messaging middleware given a topic name as a key; a service providing agent before the publishing operation and a HandlerAgent during a subscription operation. The messaging middleware returns a reference to the topic that hosts are interested in. Thus, a meeting point for those hosts is constructed.

Fig. 2 shows this system in more detail.

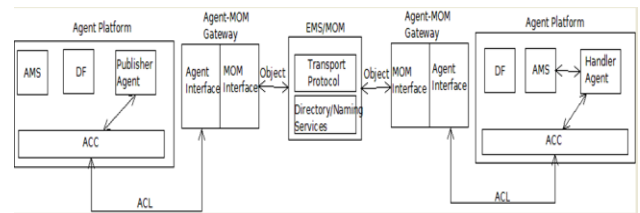


Fig. 2. System details.

This architecture relies on programming languages like java that highly supports code to be transferred between distinct machines on the network. The presented architecture achieves topic based inter platform migration between hosts of the same type of agency like JADE without any modification or between hosts of different agencies like JADE – JAM (two different Java based agent platforms), that support the same programming language with minor modifications in the application design for the APIs that the platform provides.

Service providing agent is called publisher agent as it implies its role explicitly. Supporting standards to achieve inter platform agent migration or even agent communication FIPA ACC (Agent Communication Channel) must be used. Also to achieve inter platform or even inter host communication, ACL messages should be used. Considering these points, the presented architecture offers wrapping publisher agents in ACL messages as payloads and getting these agents out of the platform or host by using ACC channel as described in FIPA specifications. On the other hand, a standard way of making computer systems communicate on a network in a loosely coupled manner with enterprise-wide standards is using Enterprise Messaging System (EMS). This is valid not only for agent systems but also for legacy software systems and agent to legacy software systems. Thus, the architecture combines all these industry standards to achieve agent migration in a highly standard way. Agent-MOM (message oriented middleware) Gateway is

considered to integrate agent systems and enterprise systems. This gateway simply transforms agents into objects that an EMS/MOM system can interpret. Gateway communicates with EMS/MOM via standard protocol like TCP or language specific standard protocol like RMI. In this way, publishing an agent wrapped in an ACL message to the topic of the EMS/MOM is achieved. Message oriented middleware notifies a subscribed entity by sending the object on its topic to MOM-Agent gateway by TCP, RMI, ...etc. communication protocol. Gateway again transforms the object into an ACL message and contacts to HandlerAgent on the agency through ACC. The HandlerAgent communicates with AMS of local agency to start the incoming agent.

Runtime scene of the system is show in Fig. 3.

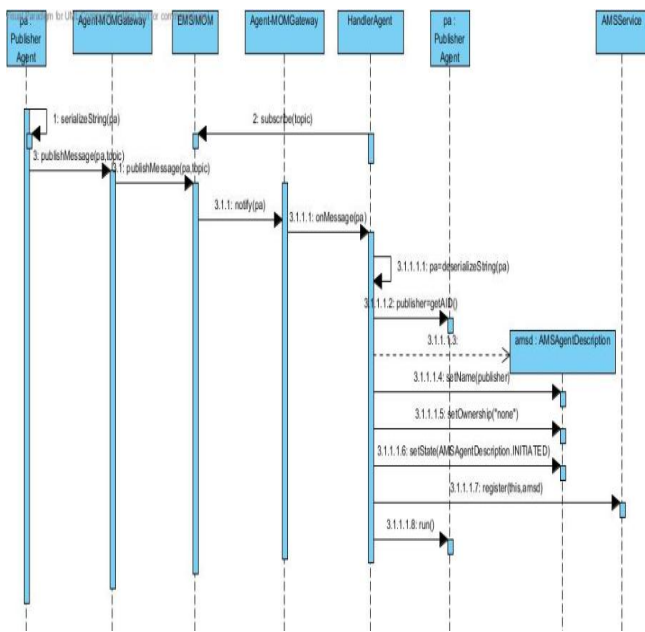


Fig. 3. Message flow between components.

Conforming to the agent concept, we need to construct behavior(s) for the agent tasks. For this reason we need to design a behavior to cope with publish primitive of publish subscribe paradigm for the publisher agent. We call this behavior as PublishCommand. PublishCommand behavior must be added to agent’s behavior list. With this behavior added to the agent, agent gains capabilities of serializing and publishing itself to EMS/MOM topic through Agent-MOM gateway. PublisherAgent (PA) delegates publish responsibility to the gateway to handle protocol conversion to make the message understandable by EMS/MOM. HandlerAgent subscribes to EMS/MOM topic through agent-mom gateway for the same reason. EMS/MOM notifies subscribed HandlerAgent with PA in a message again through gateway. HandlerAgent deserializes incoming agent to form it back and collects some information from incoming agent needed for white page services. HandlerAgent constructs descriptors for published agent to register with the AMS (Agent Management Service) of the local agency. After registering published agent with the local agency, HandlerAgent starts it. Published agent starts to work then. As a standard agent communication, publisher and subscriber agents communicate over FIPA subscribe protocol which is an interaction protocol. Given architecture

does not support direct communication between publishers and subscribers. Realization of this protocol includes EMS/MOM components and all conversions defined above. This scheme achieves not only inter platform agent migration per topic basis but also achieves agent-enterprise software communication in a standard way per topic basis.

V. IMPLEMENTATION

In this work we have implemented a prototype to see how an agent migrate from one host to another via publish/subscribe paradigm. We preferred java as an implementation language due to its advantages over mobile code. JADE (Java Agent Development framework) as an agent platform is preferred due to its fame on researchers on this topic and also because it is a java based agent framework. Fig. 4 shows layered structure. Achieving a loosely coupled communication with a messaging service, JMS (Java Messaging Service) is used for this purpose. One another thing what we need here is a gateway to integrate JADE and JMS. JMS as a new MTP (Message Transport Protocol) is integrated to JADE platform [8] by Currie et al. We chose Open JMS as a JMS provider which is supported by jade-jms gateway. In this work we proposed a topic based agent migration scheme via publish/subscribe paradigm. Therefore we proposed a solution including JADE platform for agent implementation and services, JMS as a messaging protocol and jade-jms gateway to integrate JMS as an MTP to JADE platform. This solution is composed of migrating JADE agents via JMS publish/subscribe model within JADE platform and inter JADE platforms. In our implementation we wrap JADE agents in FIPA ACL messages as payloads. Agent serializes and adds itself as a serialized payload to ACL message. Agent bearing ACL message is published to Open JMS topics created before via jade-jms gateway. Open JMS notifies subscribed target hosts via agent bearing ACL message. Subscription to JMS provider topics and handling notifications are managed by HandlerAgent component on target host. HandlerAgent design includes JMS provider topic subscriptions via jade-jms gateway and picking migrated agents from ACL messages, registering them to AMS (Agent Management System) if it is an inter platform migration and starting migrated agent. After starting, migrated agent works autonomously and terminates on the target host when it is done.

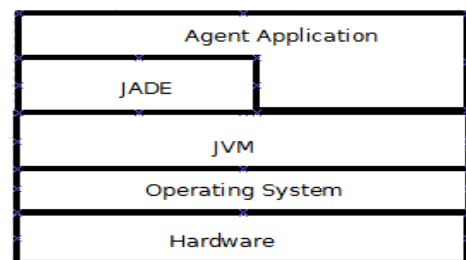


Fig. 4. Layered structure.

VI. CONCLUSION

We propose a scalable, decoupled, efficient and flexible multi agent ecosystem with service oriented architecture

given mobile agents that achieves topic based agent migration. We call this system, Information Expert Ecosystem which is a persistent system and resultant of interactive relations of information expert agents and their environments (hosts) that provide execution runtime. Information expert agents are agents that have information on a specific subject and have ability to solve problems related to the subject and can consult on that subject. Source host is a host or a subsystem that generates information expert agents and target host is a host or a subsystem that needs the help/service of the information expert agent. Information expert ecosystem is composed of information expert agents that have the ability to serve on a specific subject, source hosts that generate information expert agents and target hosts that need service or consultancy on specific subject. This ecosystem brings together service seeker hosts and service producer hosts. Thus hosts or subsystems that need each other come together without knowing each other in advance through this ecosystem. Also topic based publish/subscribe scheme is adopted for agent migration which differs than existing agent migration schemes and achieves inter platform agent mobility.

REFERENCES

- [1] J. Ametller, S. Robles, and J. Borrel, "Agent Migration over FIPA ACL Messages," *Mobile Agents for Telecommunication Applications Lecture Notes in Computer Science*, vol. 2881, pp. 210-219, 2003.
- [2] S. Malik, N. Qureshi, A. Ali, H. Ahmad, and H. Suguri, "Inter Platform Agent Mobility in FIPA Compliant Multi-Agent Systems," *IEEE International Conference on Active Media Technology*, 2005, pp. 393-396.
- [3] M. Paraschiv, A. Stefanescu, and A. Almasi, "Agent Based Implementation of a P2P Publish/Subscribe System," *Computer Science Master Research*, 2011.
- [4] Z. Shen, R. Li, and J. Luo, "Mobile Agent Based Middleware using Publish/Subscribe Mechanism in Wireless Sensor Networks," *International Conference on Communication Software and Networks*, 2009.
- [5] O. K. Sahingoz and N. Erdogan, "AGVENT: Agent Based Distributed Event System," in *Proc. 30th The Conference on Current Trends in*

Theory and Practice of Computer Science, Prag, Czech Republic, 2004.

- [6] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The many faces of Publish/Subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114-131, June 2003.
- [7] V. Pham and A. Karmouch, "Mobile Software Agents: an Overview," *IEEE Communications Magazine*, 1998.
- [8] E. Curry, D. Chambers, and G. Lyons, "A JMS Message Transport Protocol for the JADE Platform," *IEEE International Conference on Intelligent Agent Technology*, 2003, pp. 596-600.



Mustafa Akif Karzan was born in Istanbul, Turkey in 1983. He has BSc. degree in Electronics and Communications Engineering field from Faculty of Electrical and Electronics Engineering in Istanbul Technical University, Istanbul, Turkey since 2008. He works on multi-agent systems and specifically on agent communication and migration architectures.

He is currently MSc. Student in Computer Science field in the Institute of Informatics in Istanbul Technical University, Istanbul, Turkey.



Nadia Erdogan obtained her BSc. degree in Electrical Engineering, Computer branch from Bosphorus University, Istanbul, Turkey since 1978. She has MSc. Degree in Computer Science from Bosphorus University. She has doctoral degree from Institute of Science in Istanbul Technical University. She works on distributed systems, agents systems and parallel programming fields.

She is working as an instructor at Computer Engineering department of Computer Engineering and Informatics Faculty in Istanbul Technical University in Istanbul, Turkey. She has involved in writing the book "A Java-Based DSM System for User Defined Shared Data Objects", Şahingöz Ö. K., Erdoğan N., Software and Hardware Engineering for the 21th Century, Editor: N.E. Mastorakis, World Scientific and Engineering Society Press (WSES_Press), ISBN:960-8052-06-8, 1999. Her publications include, "Logistics Application on Distributed Composite Object Based Environment", Nakhchivanski J, Yılmaz G., Erdoğan N., 4th. International Conference on Computer and Electrical Engineering, Singapore (ICCEE 2011), October 14-15, 2011, "Hierarchical, Searchable, Secure, Persistent Publish / Subscribe Software Framework", Osanmaz Çelik, P., Erdogan, N. EURO Working Group on Decision Support Systems - Workshop on Decision Systems, (EWG-DSS London-2011), London, UK, June 23-24, 2011.