International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(2): 412-427
The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

# Agent-based MOM Interoperability framework for integration and communication of different SOA applications

Najhan M.Ibrahim, Mohd Fadzil Hassan, Zain Balfagih
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
hanfast@gmail.com, mfadzil_hassan@petronas.com.my, z_balfagih@hotmail.com

## ABSTRACT

A lot of research works in different areas have been done on Interoperability frameworks. Each area of research has its own specification and requirements to manage a complex processing request. Interoperability framework is significantly important for utilizing heterogeneous communications and integrations in SOA systems. Various middleware technologies are often used as the communication and integration infrastructure for distributed system to ensure the interoperability. Nevertheless, different types of middleware have raise interoperability issues as well. One of the well known middleware architectures used for heterogeneous systems is Message Oriented Middleware (MOM). MOM is known as common attach technology for heterogeneous, distributed and loosely coupled software system. In this paper we explore and analyze studies related to cross-platform in communication and integration. In the end, we have proposed agent-based MOM interoperability framework and Ontology mapping conceptual model which is the key model to do message mapping for different applications in SOA system.

## KEYWORDS

Service Oriented Architecture (SOA), Message Oriented Middleware (MOM), Agent Technology, Web Services, Interoperability, Ontology mapping.

## 1 INTRODUCTION

Service-oriented architecture (SOA) aims to interconnect distributed, loosely coupled, and interoperable components of software owned or provided by different domains. For example, many applications and heterogeneous platforms require a process flow of communication to solve interoperability problem in cross-platform systems [1]. Thus, insuring an interoperable communication between cross-platform systems over Internet is the main problem for Service Oriented Architecture. A suitable Interoperability framework would enable these requirements is required.

In addition, Interoperability framework is an architecture where a composite of processes is implemented as an interconnected queue of smaller, less complicated tasks [2]. The concept of interoperability framework has been successfully implemented in many areas of research. Due to the great capabilities of interoperability framework to glue the system components, it has been emerged for the SOA systems along with the agent-based technology. Currently, interoperability frameworks across multi domain of systems are evoking a high degree of interest in research. In this paper we describe our approach that utilize agent technology to build MOM based interoperability framework [3].

Furthermore, the fast growing of SOA architecture with Web Services and other connectivity mechanisms such as SOAP and REST are clearly important for system developments. Moreover, the loosely coupled nature of the message queuing (MQ) model, a large number of existing MQ customers feel that they are already adopting SOA principles. MQ provide an asynchronous communications protocol, meaning that the sender and receiver of the message do not need to interact with the message queue at the same time. The MQ Service allows MQ applications to be catalogued as software assets which can then be reused and composed as Web Services. As a communication channel in the distributed computing environments like SOA, MOM normally provides a message queue between the service provider and the service consumer to transfer messages. MOM is typically asynchronous, but it also supports synchronous message passing as well [MOM]. MOM is the foundation of Enterprise Service Bus (ESB) [ESB], which is widely believed to be a cornerstone of many SOA environments.

According to our literature review, we realized that the limitation of MOM architecture still existed. Regarding to the extension of Message Oriented Middleware (MOM) for integration and communication in distributed system, some of these research works are problem specific and some are too general and not identify any particular issues. Several studies conducted to solve different interoperability communication in each specific issue such as congestion control in a reliable Scalable, enabling non-repudiation and composite-event-based [4-6]. Generally, MOM provides a high performance foundation for application interoperability in diverse and complex environments. At the same time, Agent-based technology is one of the well-known software technologies for cross-platform communication. It includes significant attributes such as autonomous, adaptive, interactive, support multi-protocol and lightweight implementation of atomic reaction [3]. In addition, a lot of research works has been conducted in the literature review regarding the enhancement of agent-based technology in cross-platform communication [7].

Therefore, in our interoperability framework, we will combine these two technologies for cross-platform communication within SOA systems. The remaining of this paper is organized as the following. In section 2 and section 3, we present the motivation of study and research problem. In section 4 we briefly discuss background of the research with the implemented methodology. In section 5 we present as discuss the comparative study as related work and in section 6 we present proposed our communication workflow. In section 7 we present our suggested framework and the last section, we conclude the paper and discuss the possible future work.

## 2 MOTIVATIONS

There are four common challenges in adopting SOA systems which also motivate us to study the different style of SOA implementation. First challenge is managing services, lack of testing in SOA space, appropriate level of security and the last is interoperability. Interoperability became an important aspect of SOA implementations since different vendors have their own product and deployment style to implement SOA system. Therefore, integration and communication between those products will be interrupted. In addition, interoperability can be defined as Information Technology ability to work with each other. In the loosely coupled environment of SOA, separate resources don't need to know the details of each work, but they need to have enough common ground to exchange messages without error or misunderstanding [8]. In this paper, interoperability refers to the use of information system as tools to assure services or messages exchange between heterogeneous and geographically distributed within organizations in Service Oriented Architecture (SOA).

Recently, researchers have shown an increased interest in interoperability since there are a lot of products available to provide the same functionality which have raised some heterogeneous issues [9]. Various middleware technologies are often used as the communication infrastructure for distributed system to ensure the interoperability. However, different types of middleware, deployment style and even version of product have cause interoperability issues as well. We believe that more research needs to be explored and consolidate for interoperability issues. Development of generic, flexible and autonomous interoperability framework would be a good step towards achieving interoperability in the SOA systems. Traditionally, researchers within this area are focusing more on the interoperability specification, standardization and requirement [10].

The motivation for this research work is to investigate:

1. How to develop a generic, flexible and autonomous interoperability framework for cross-platform integration and communication in SOA system that can support multi-type of message in different SOA application.

2. How to adopt Message Oriented Middleware architecture with agent-based technology for an intelligent, flexible and generic interoperability framework.

3. How to develop an automated multi agent system for message translation and communication across multi domain of SOA applications.

4. How to develop interoperability framework for SOA environment that support multi domain of applications.

## 3 RESEARCH PROBLEMS

There are many on going research works on different areas of interoperability such as semantic interoperability, scientific workflow interoperability, network collaboration interoperability and so on [8]. Difference interoperability frameworks would be in different styles of development and infrastructure. Most of studies are problem specific to solve only their interoperability issue. Those interoperability frameworks will be best interoperate within their organization framework and infrastructure but it would create interoperability barrier with others type of interoperability framework [1, 11].

As a result, we are strongly believed that more research study is needed in this area to support current complex system architecture.

In this paper, we conducted a comparative study between several interoperability frameworks with consideration of theoretical and conceptual framework toward cross-platform interoperability framework for SOA environment. First, we performed a comparative study to evaluate the significant of all interoperability attributes for cross-platform integration and communication. Then, we have selected most significant interoperability attributes to be deployed in our interoperability framework. Based on our literature study, we found that many works has been conducted to solve interoperability issues but in term of generic, flexible and autonomic still very lack of consideration. Bellows are some common attributes which still very less consideration in existing interoperability framework.

1. Most of existing frameworks still do not fully support asynchronze type of communication.
2. Lack of system failure recovery.
3. Do not support guaranteed transmission.
4. Both system need to be online for any process or transmission.
5. Unable to integrate and communicate across multi domain of SOA system.
6. Most of frameworks are manual process, very lack in autonomic execution by system applications.

## 4 BACKGROUND AND METHODOLOGY

Agent-based infrastructures can, normally, facilitate the autonomous communication between distributed and loosely coupled systems. Agent-based technology is a communication architecture that provides interoperability advantage for the cross-platform systems [3]. Currently, different enhancements have been implemented on agent-based communication for different purposes. These enhancements can overcome some interoperability problem but having different kind of technologies and development styles can cause some interoperability issues. As the software architecture of any organization system is the significant part of the organizations' systems, some requirements needed to be included in the current communication workflow such as 24 by 7 environment, autonomous level, failure recovery and guaranteed transmission. Therefore, existing agent-based technologies still have some barrier of communication on each others [12, 13].

Constructing of flexible, autonomous and generic communication workflow would be a good initial step to achieve interoperability in the SOA system. Currently, researchers within this area are focusing more on the interoperability specification, standardization and requirements. Furthermore, agent-based technology can be useful tool to facilitate cross-platform communication which provides some significant attributes such as auto mouse, adaptive, interactive, support multi-protocol and lightweight implementation of atomic reaction [14].

To evaluate the efficiency of cross-platform communication between different distributed systems and to prove the capability of such communication, generic specifications of the requirements need to be identified. It is also necessary to explore some metrics to measure and evaluate the interoperability. In the context of SOA interoperability [4], agent-based technology and message oriented middleware (MOM) would be a great combination to develop communication platform for cross-platform infrastructure [15].

In this study, general literature review has been conducted. Based on our examination, different extensions of agent-based technology have focused in different kinds of requirements for cross communication. In the related work section, we present a table that summarize the literature study regarding several attributes of different agent-based deployment for distributed communication where every approach will be discussed in detail in the next section.

In order to achieve the objective of research work, we used different methods throughout our research work. We have divided our work to 3 different phases and in this section, we show the different methods and activities used for every phases of this research as the following:

Phase 1: Analyzed and examined interoperability requirements for cross-platform communication in SOA applications.

- Conducted literature survey to understand about interoperability requirements and properties for SOA environment. The different classifications for these attributes have been considered such as theoretical, conceptual, standard and interoperability requirements.
- Examined the existing interoperability framework for distributed and cross-platform within SOA context.
- Conducted a comparative study for existing interoperability framework within SOA context to find the weakness and advantages of each framework to be considered as quality attribute in our interoperability framework.
- Analyze each attributes properties of interoperability requirements to be included in our interoperability framework.
- The main attributes of interoperability framework have been identified to build a conceptual interoperability framework.

Phase 2: Developing a generic, autonomic and flexible interoperability framework.

- Developing a combination of Message Oriented Middleware archiecture with a Multi Agent System (MAS) for interoperability framework based on phase 1 analyzed and examined.

Phase 3: Prototyping and Simulate the proposed framework

- Prototyping the proposed system using NetBean and JADE environment and using Web Service Integration Gateway (WSIG) to communicate with Web services.
- Evaluate the framework accuracy and reliability of communication and compare with some existing traditional frameworks.

## 5 RELATED WORKS

Most of Message Oriented Middleware (MOM) is implemented with queued message and store-and-forward capability which known as Message Queuing Middleware (MQM). In general, MOM provides a high performance foundation for application interoperability in diverse and complex environments. At the same time, Agent-based technology is one of well-known software technology for cross-platform communication [3]. It also includes some of significant attributes such as autonomous, adaptive, interactive, support multi-protocol and lightweight implementation of atomic reaction. Therefore, in our proposed communication workflow, we will combine these two technologies for cross-platform communication.

Many research studies have been found in literature regarding the enhancement of agent-based technology in cross-platform communication. Most of research works are problem specific and some are too general and not specific for one particular issue [7]. Several research works have been conducted to solve different issues of communication such as reliable asynchronous of communication and form-based agent communication language. As shown in Table 1, 12 research works relevant to agent-based extensions derived from literature study regarding distributed communication of SOA are presented.

Table 1 summarizes and compares different enhancements of agent-based technology with regards to 8 proposed attributes concerning interoperability as highlighted in the first row of the table. We can split these extensions into 2 categories; synchronize and asynchronize communication. The enhancements were selected from several kinds of technologies, in which some of them are very basic using of agent-based technology but they consist of important highlighted points for interoperability of heterogeneous system. The attributes used in the evaluation were essentially selected from generic specification requirement for cross-platform communication and also from seamless interoperability guidelines [1].

The definition of each attributes included in Table 1 can be summarized as follow [1].

1) Communication type is the type of communication style.

2) 24 by 7 environments can be defined as the availability of application to process others request.

3) Autonomous level refers to the intelligent level of system to manage and implement the request of other systems.

4) Message type means the type of message that used in communication such as SOAP and XML.

5) Software failure recovery is the ability of application itself to recover the failure of transmission.

6) Guaranteed transmission can refer as high levels of application transmit the message to the partner.

7) Scalability is the ability to handle number of requests and number of servants.

8) Follow specifications mean the level of standardization applied to develop the architecture.

**Table 1. Agent technology for cross-platform comparison table**

| No | Technology using for cross-platform communication | Communication type | Availability and 24x7 support | Autonomous level | Message type | Software failure recovery | Guaranteed transmission | Scalability | Follow Specifications |
|---|---|---|---|---|---|---|---|---|---|
| 1 | An Agent Platform for Reliable Asynchronous Distributed [7] | Asynchronous | Medium | High | ACL | Medium | Low | Low | Medium |
| 2 | Agent-Based Middleware for Web Service Dynamic [11] | Synchronize | Low | High | WSDI | Low | Low | Medium | Medium |
| 3 | XML-based Mobile Agents [6] | Synchronize | Low | High | XML | Low | Low | Low | High |
| 4 | An Agent-Based Distributed Smart Machine [5] | Synchronize | Low | High | KQML / ACL | Medium | Low | Medium | High |
| 5 | An Agent XML based Information Integration Platform [12] | Synchronize | Low | High | SOAP | Low | Low | Low | Medium |
| 6 | A Cross-Platform Agent-based Implementation [13] | Synchronize | Low | High | ACL | High | Low | High | Medium |
| 7 | Communication System among Heterogeneous Multi -Agent System [14] | Synchronize | Low | High | ACL | Low | Medium | Medium | High |
| 8 | FACL (Form-based ACL) [15] | Synchronize | Medium | High | Form-based (ACL) | Low | Medium | Low | Low |
| 9 | ACL Based Agent Communications in Plant Automation [16] | Asynchronous | Medium | High | ACL | Low | Medium | Medium | High |
| 10 | Multi-agent Systems for Distributed environment [17] | Synchronize | Low | High | ACL / KQML | Low | High | Medium | High |
| 11 | SOA Compliant FIPA Agent Communication Language [10] | Synchronize | Low | High | ACL | Low | Low | Medium | High |
| 12 | An Agent-Based Distributed Information Systems Architecture [5] | Synchronize | Low | High | ACL | Low | Medium | High | Medium |

L. Bellissard , N. De Palma, A. Freyssinet, M. Herrmann and S. Lacourte [14] introduce a distributed communication model based on autonomous by agents software. Agents act as the glue software components and they provide atomic execution or migration from node to node. A. Lin and P. Maheshwari [16] aims to construct an agent-based middleware for Web service dynamic integration on Peer-to-Peer networks to facilitate the integration of optimal quality of Web services for application integration. R. Steele T. Dillon P. Pandya and Y. Ventsov [13] presents a mobile agent system design based on the use of XML-based agent, the UDDI registry for agent registration and lookup/discovery and XML Web Service calls for mobile agent intercommunication and migration. In these works, the highlighted techniques or requirements are too basic for interoperability communication of heterogeneous distributed information systems. Many attributes were not considered as suggested in interoperability Table 1, such as 24 by 7 environment, software failure recovery, guaranteed transmission and scalability.

Y.C. Kao and M.S. Chen utilized [17] software agent based technology in enhancing remote tool service system by develop related remote service ontology to manage its smart and distributive characteristics. X. Li [16] mentions that to integrate process operation systems effectively, an Agent/XML based information integration platform is proposed. The subsystems are encapsulated as Agents-based on XML and Agent technology. Then based on integration goal of, the Agents distributed in different domains are integrated. The encapsulation of different subsystems was implemented through Agent technology. E. Cervera [18] presents a cross-platform, networked implementation of control tasks based on software agents and streaming technologies.This implementation consists of free, off-the-shelf, software components, resulting in a transparent system whose configuration can be adapted to existing hardware in very different ways, without any modification in the source code. However, according to Table 1 these papers still lack of some generic communication requirement such as scalability, guaranteed transmission, Software failure recovery and it also does not support 24 by 7 environments which is very important for current software architecture.

Y. Yoon, K. Choi and D. Shin [19] suggest and execute the system for translation among heterogeneous service description languages. Any of multi-agent system use the proposed system can expand the service area. T. CHUSHO and K. FUJIWARA [16] describe a multi-agent framework and an agent communication language (ACL) for the MOON (mu1tiagent-oriented office network) systems which are distributed systems of E- commerce. The multi-agent framework is a Java application framework and includes a form-based ACL (FACL) as a common protocol for passing application forms. Q. Feng and G. Lu [6] have tried to use agent communication language (ACL) to implement communications between PABADIS agents. In order to fulfill with the final part of project in which Grasshopper is used as agent developing platform, FIPA ACL is adopted. In contrast, these in papers did not discuss several attributes which are available in other distributed communication architectures such as 24 by 7 environments, software failure recovery, guaranteed of transmissions and Scalability. Therefore, it would be very difficult for this framework to interact with other systems especially if those systems were built based on other interoperability requirements [10].

H. Farooq Ahmad [20] temporarily explores the basic agent systems architecture with highlight on agent communication languages (ACL). Two most accepted agent communication languages, namely FIPA ACL and KQML have been briefly reviewed. This work summarizes generic requirements of the agent system from the viewpoint of FIPA and OMG models. M.

AtifNazir, H. Ahmad and H. Suguri [7] address the issue of interoperability between FIPA agents and grid services by suggesting a communication framework for interconnected. SOA compliant FIPA ACL ontology has developed that provides the necessary expressions that not only facilitate to the FIPA specifications but also used by grid services. M. Purvis, S. Cranefield, G. Bush and D. Carter [21] describes framework for building distributed information systems from available resources, based on software agent and distributed technologies. An agent-based architecture is adopted, by messages exchanging via the FIPA agent communication language (ACL). Similar to other research work these efforts did not include some significant interoperability communication attributes such as 24 by 7 environment, software failure recovery, guaranteed transmission, and scalability. Therefore, this study strongly proposes cross-platform interoperability attributes as a based guideline for future cross-platform communication framework.

## 6 AGENT-BASED MOM WORKFLOW MANAGEMENT SYSTEM.

In agent technology, agent the interactions between agents via agent communication languages (ACL) which is the basic standard agent communication proposed by FIPA (Foundation for intelligent Physical Agent). Workflow system can coordinate or control the interactions between agents which used to perform tasks, such as message passing and computing tasks. Our approach for workflow management in SOA environments is agent-based MOM Flow Management Systems (AMFMS). AMFMS provides high-level middleware to enable transparent communication between agents over distributed system and resource sharing over a wide-area network.

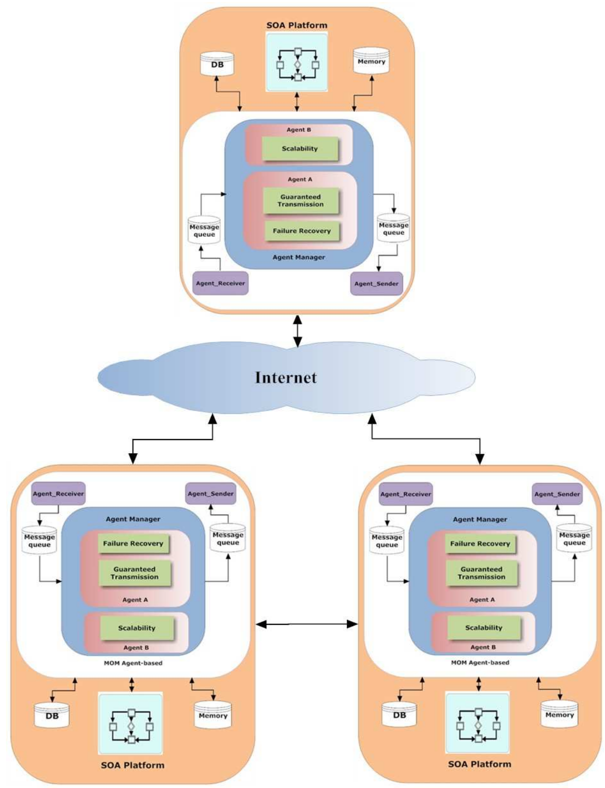


Fig. 1. The Architecture of AMFMS

The basic ideas for the AMFMS are very simple: it consists of a collection of federated servers with hosting AMFMS engine in each of SOA system. The partnership of processing resources, which host the AMFMS environment, make their own placement and communication decisions independent of each other. The AMFMS environment provides the necessary framework for the seamless communication and execution of the component parts of the users' requests across the distributed cross-platform system to ensure that the request is fulfilled.

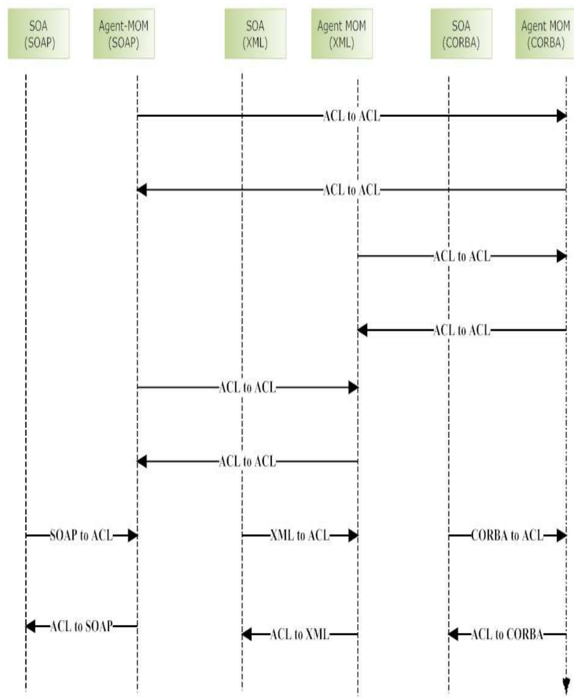


Fig. 2 Agents Communication Activities

AMFMS architecture has been adopted from a service-oriented perspective, with a high degree of automation that supports flexible collaborations and computations on a large complex application, as shown in Fig.1. In the architecture, workflow applications are central to the architecture of the AMFMS. Workflow engines are distributed across SOA environment.

We adopted the cMOM (Composite-event Based Message-Oriented Middleware) [4] as AMFMS engine. Communication or message passing can manage themselves in one or more workflow engines. AMFMS engine can be interconnected with those SOA services and resources in the engine. AMFMS engines can be dynamically detected and connected into different SOA architectures, such as XML-based architecture, SOAP-based architecture or REST-based architecture. Due to the dynamic nature and flexibility of agent-based technology, the AMFMS is suitable for future communication workflow.

## 7. SUGGESTED FRAMEWORKS

Most of current discussions in distributed system is middleware, as an interoperate application because of its properties in cost and time savings (increases productivity) [22]. Even though web services over SOAP are the most accepted communications mechanism for SOA, other communication techniques are used such as REST, and messaging services [23]. Nevertheless, organization cannot truly take advantage of an application’s business benefits without a well-integrated and communicate of distributed software infrastructure. Middleware enables this integration, approaching applications out to distributed environments and release the domain-specific value of each application.

One of the main techniques used to solve integration problems between distributed systems is Message Oriented Middleware (MOM). MOM is becoming increasingly difficult to ignore in integration of distributed geographical systems. It is one of three categories of communication middleware that provide program-to-program connection by message passing. Generally, it supports multiple protocols that consist of facility to support reliable and scalable high-performance distributed platforms. Most of MOM environments are implemented with queued message store-and-forward capability, which is Message Queuing Middleware (MQM). The two other categories of connectivity middleware platform are Remote Procedure Calls

(RPC) and conversational programming such as Common Programming Interface for Communications (CPI-C). In general, MOM provides most high performance foundation for application interoperability in diverse and complex environments [3, 24, 25]. For these reasons we suggest to use MOM technology to solve the interoperability problem between SOA systems.

However, in this paper we explored and analyzed studies related to cross-platform integration and communication. We found that there are some weaknesses of MOM which are [4].

• 1) Able to support only a small number of execution and limited the ability to link with complex applications.
• 2) Unable to trace the execution of the application and cannot rollback tasks at any level. The execution models of MOMs based on are too simple. Therefore, it cannot record any level of execution in applications and maintain the states of those executing tasks.
• 3) When users need to terminate tasks, MOM cannot correctly rollback all or some specified tasks.
• 4) Limited ability to link complex application.
• 5) Losing the trace of execution.
• 6) Unable to rollback the task.
• 7) Low level of autonomous.

To overcome these weaknesses, some researches have been conducted to extend MOM architecture. Most of these efforts are addressing one or two of the MOM limitations and not the general requirement of cross-platform interoperate. Therefore, we strongly argue that research in this area is needed to consolidate these distinct interoperability cross-platform integration and communications to find a common ground toward the establishment of a generic framework for interoperability cross-platform communication to be applied within the SOA context.

Development of generic and flexible framework, architectures, tools and platforms to support interoperability among heterogeneous and geographically distributed organizations would be a good step towards achieving interoperability in the SOA systems. Most of researchers within this area are focusing more on the interoperability specification, standardization and requirement [10, 11]. Based on [11] there are 27 interoperability requirements to achieve collaborative and integration. After we have conducted literature survey, we found that there are some attributes are overlap and some attributes still not take into consider. Therefore, we have proposed 16 interoperability requirements for cross platform integration and communication [1]. In this research work, we will focus on 8 most significant interoperability requirements. It would be difficult to implement all 16 attributes in our first pilot study. We have conducted generalization for all interoperability requirements, according to some theoretical and conceptual of seamless interoperability. Generally, 8 basic interoperability requirements are frequently adopted for cross-platform and distributed system as following description [26].

1) Communication type is the type of communication style.

2) 24 by 7 environments can be defined as the availability of application to others processes the request.

3) Autonomous level refers to the intelligent level of application to manage and implement the request of other applications.

4) Message type means the type of message that used in communication such as SOAP and REST.

5) Software failure recovery is the ability of application itself to recover the failure of transmission.

6) Guaranteed transmission refers to the high levels of application ability to transmit the message to the partner.
7) Scalability is the ability to handle number of requests and number of servants.

8) Follow specifications mean the level of standardization applied to develop the architecture.

Many works have been found in literature regarding the extension of Message Oriented Middleware (MOM) in cross-platform communication. Some of these research works are problem specific and some are too general and not specific for one particular issue. Many research works have been conducted to solve different interoperability communication in each specific issues such as congestion control in a reliable Scalable, enabling non-repudiation and composite-event-based. As shown in Figure 3, is the proposed architecture for Agent-based MOM derived from literature study regarding distributed and cross-platform communication in SOA are presented.

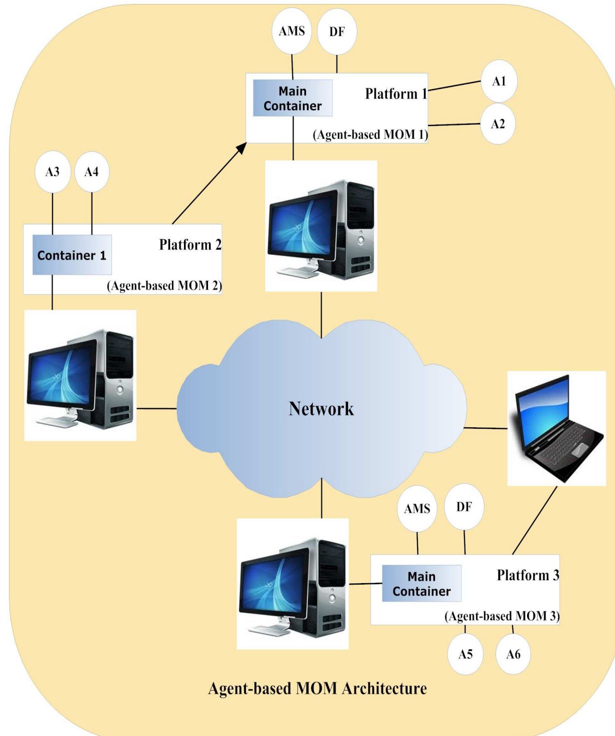


Fig.3 Agent-based MOM Architecture

Agent-based MOM (ABM) architecture has been adopted from a service-oriented perspective, with a high degree of automation that supports flexible collaborations and computations on a large complex application, as shown in Fig.3. In the architecture, workflow applications are central to the architecture of the ABM. Workflow engines are distributed across SOA environment. In agent-based MOM workflow, all components will be two way interactions and the agent-based MOM platform will use the standard messaging proposed by FIPA which is ACL (Agent Communication Language), each agent will analyze each message priorities and in which action should be taken. For example, SOAP platform need to send message or run some service in REST platform where analyzer agent will analyze the message that received by agent-receiver. Then, it will translate it to ACL (Agent communication Language). The sender agent also will identify which platform offer which type of service and which message type they are using [5].

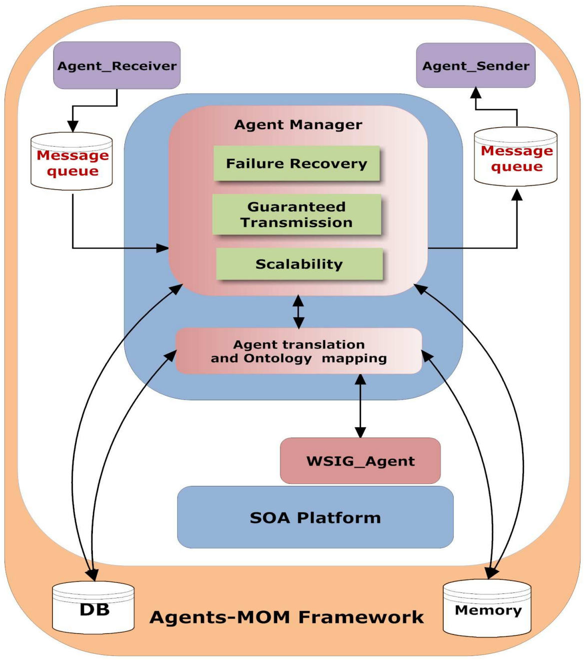


Fig. 4 Agent-based MOM Framework

This paper presents a novel framework to automate the integration and communication across multi domain of SOA systems to ensure the interoperability between different SOA applications by using agent technology. This framework uses two main technologies, Message Oriented Middleware (MOM) architecture and agent technology. MOM as a common glue technology for distributed systems and agent technology as an adaptive and flexible in nature of application , we used MOM concept of message queue with asynchronize type of communication to manage the integration and multi-agent system to handle the autonomous interaction between different applications. The interoperability architecture consists of seven components as the following:

CONCEPTUAL DESCRIPTION OF FRAMEWORK

1. In cross-platform communication each frameworks can use different type of message for their communication of SOA systems.

2. The communication between frameworks will use ACL (Agent Communication Language)

3. The sender agent in every framework will translate the message to ACL for cross system communicate.

4. The receiver agent in every framework will translate message from ACL to their own message domain such as SOAP.

5. The agent manager will manage of every send or receive of message which it will focus on failure recovery, guaranteed of transmission and ability to handle number of requests.

6. Message queue will provide as storage queue for sending and receiving of message.

7. Agent translation with Ontology mapping consist of following attributes
7.1 Agent Description
7.2 Ontology description
7.3 Message type

7.1 The Suggested Ontology Mapping

[4] Has highlighted some limitation of MOM such as the lack of ability to link complex application, lose trace of execution and autonomous. A research effort on Composite event-based has

resolved some issues on combination of composite events. However, according to some survey paper there are some specifications which are lake of consideration for interoperability cross-platform communication such as 24 by 7 environments, autonomous level, software failure recovery and scalability [28-31].

This paper focuses on communication-level with high level of autonomous and ontology mapping. The goal is to define the most promising agent-based MOM model to support interoperability cross-platform communication framework. One advantage of our approach is that message or services can be managed dynamically where agent-based and ontology mapping will be the keys concept. This is especially suitable for heterogeneous SOA environments [32].

The suggested framework (Fig.5) has been enhanced from agent-based MOM. Agent-based Message Oriented Middleware developed to enables cross-platform communication in SOA system. In addition, the level of automation in current transaction or communication became significant. Therefore, in the conceptual Ontology Mapping has applied the concept of basic ontology mapping. Furthermore, our research works as shown in Fig 5 puts the initiatives to improve the level of automation and ontology mapping with similarity of service description for the better of future cross-platform communication architecture [26, 33, 34].

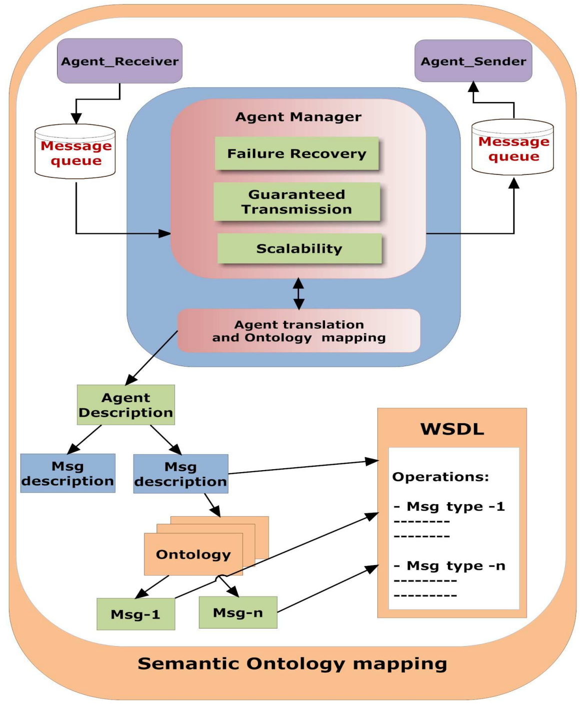

Fig.5 Conceptual Ontology Mapping

In addition, this communication workflow decreases the level of complexity across multi domain of SOA application. Classically, SOA architecture needs to deal with middle server such as CA and message brokers. According to [29] SOA system is not required to have any third party in the system between trusted domains. By decreasing such complexity of workflow, it will increase the autonomous level that doesn't need any responds from participant. In agent-based MOM interoperability framework, all components will use asynchronize type of communication to interaction and the massaging platform will use the standard messaging proposed by FIPA which is ACL (Agent Communication Language), each agent will analyze each message priorities and in which action should be taken. For example, in sender side, it gets message from SOA system then translate into ACL format that will enable the cross-platform communication. For receiver, after it receives ACL message then will perform a message mapping which need to translate to the right message type of their SOA application such as if the SOA system support REST then the agent translation will translate from ACL to REST format [35].

## 8 CONCLUSION & FUTURE WORK

Cross-platform communication became a significant facility for connecting and sharing resources between SOA-based applications. The lack of interoperability awareness in SOA inspires researchers to focus on heterogeneous type of communication framework between SOA systems. We found that some researchers focus on some particular perspective while others have worked to combine several specifications such as scalability, flexibility and resource management. We believe that more research in this area is still needed.

The goal of this research is to provide an autonomic and interoperability communication between cross-platform of SOA system. This is to ensure the exchange of message or resources without any boundaries. Furthermore, Agent-based MOM Interoperability framework is proposed. Agent-based MOM enables the communication between different SOA-based applications and also for resource sharing across organizations. In Agent-base MOM, we propose a communication based on agent communication language (ACL) which able to communicate over cross platform. This is to guarantee the exchange of data without any restrictions. The proposed agent-based MOM is for cross-platform environment within SOA system. In the future, the main concern of this research will concentrate on supporting more different type of SOA-based applications to construct more valuable interoperability framework across multi domain of SOA systems.

## 9 REFERENCES

1. M.Ibrahim, N.b. and M.F.b. Hassan, A Survey on Different Interoperability frameworks of SOA Systems Towards Seamless Interoperability, in ITsim '10. 2010, IEEE: Kuala Lumpur, KLCC.
2. Yuan, P., et al., Research on an MOM-Based Service Flow Management System, in GCC 2004. 2004, Springer: Verlag Berlin Heidelberg.
3. Goel, S., H. Shada, and D. Taniar, Asynchronous Messaging Using Message-Oriented-Middleware, in IDEAL 2003. 2003, Springer: Verlag Berlin Heidelberg. p. 1118-1122.
4. Yuan, P. and H. Jin, A Composite-Event-Based Message-Oriented Middleware, in GCC 2003, LNCS 3032. 2004, Springer: Verlag Berlin Heidelberg. p. 700–707.
5. Pietzuch, P.R. and S. Bhola, Congestion Control in a Reliable Scalable Message-Oriented Middleware, in International Federation for Information Processing. 2003, IFIP USA. p. 202–221.
6. Parkin, S., D. Ingham, and G. Morgan, A Message Oriented Middleware Solution Enabling Non-repudiation Evidence Generation for Reliable Web Services, in ISAS 2007. 2007, Springer: Verlag Berlin Heidelberg p. 9 – 19.
7. Raja, M.A.N., H.F. Ahmad, and H. Suguri, SOA Compliant FIPA Agent Communication Language. 2008, IEEE.
8. Yanga, Q.Z. and Y. Zhangb, Semantic interoperability in building design: Methods and tools. Computer-Aided Design, 2006. 38.
9. Kuppuraju, S., A. Kumar, and G.P. Kumari, Case Study to Verify the Interoperability of a Service Oriented Architecture Stack, in International Conference on Services Computing. 2007, IEEE: India.

10. Elmroth, E., F. Hernández, and J. Tordsson, Three fundamental dimensions of scientific workflow interoperability: Model of computation, language, and execution environment. Future Generation Computer Systems, 2009.
11. Chituc, C.-M., A.r. Azevedo, and s. Toscano, A framework proposal for seamless interoperability in a collaborative networked environment. Computers in Industry, 2009: p. 22.
12. Goel, S., H. Sharda, and D. Taniar, Message-Oriented-Middleware in a Distributed Environment, in IICS 2003. 2003, Springer: Verlag Berlin Heidelberg. p. 93-103.
13. Steele, R., et al., XML-based Mobile Agents, in Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05). 2005, IEEE.
14. Bellissard, L. and A.F. N. De Palma, M. Herrmann, S. Lacourte, An Agent Platform for Reliable Asynchronous Distributed Programming. 1999, IEEE: FRANCE.
15. Lin, A. and P. Maheshwari, Agent-Based Middleware for Web Service Dynamic Integration on Peer-to-Peer Networks, in AI 2005, LNAI 3809. 2005, Springer: Verlag Berlin Heidelberg. p. 405 – 414.
16. Li, X., An Agent/XML based Information Integration Platform for Process Industry, in 2010 2nd International Conference on Computer Engineering and Technology. 2010, IEEE.
17. CHUSHO, T. and K. FUJIWARA, A Form-based Agent Communication Language for Enduser-Initiative Agent-Based Application Development. 2000, IEEE.
18. Cervera, E., A Cross-Platform Agent-based Implementation. 2005, IEEE.
19. Tai, S., T.A. Mikalsen, and I. Rouvellou, Using Message-oriented Middleware for Reliable Web Services Messaging in WES 2003, LNCS 3095. 2004, Springer: Verlag Berlin Heidelberg. p. 89–104.
20. Ahmad, H.F., Multi-agent Systems: Overview of a New Paradigm for Distributed Systems, in HASE'02. 2002, IEEE.
21. Kao, Y.-C. and M.-S. Chen, An Agent-Based Distributed Smart Machine Tool Service System, in 3CA 2010. 2010, IEEE.
22. Papazoglou, M.P., et al., Service-Oriented Computing:State of the Art and Research Challenges. 2007, IEEE.
23. Issarny, V., M. Caporuscio, and N. Georgantas, A Perspective on the Future of Middleware-based Software Engineering, in Future of Software Engineering. 2007, IEEE.
24. Banavar, G., et al., A Case for Message Oriented Middleware, in Verlag Berlin Heidelberg, Springer, Editor. 1999, Springer: Berlin p. 1–17.
25. Ahn, S. and K. Chong, A Case Study on Message-Oriented Middleware for Heterogeneous Sensor Networks, in IFIP International Federation for Information Processing. 2006, IFIP. p. 945 – 955,.
26. M.Ibrahim, N., M.F. Hassan, and Z. Balfagih, Agent-based MOM for Interoperability cross-platform communication of SOA Systems, in SHUSER2011. 2011, IEEE: Kuala lumpur.
27. Laumay, P., et al., Preserving Causality in a Scalable Message-Oriented Middleware, in Middleware 2001, LNCS 2218. 2001, Springer: VerlagBerlin Heidelberg. p. 311–328.
28. Choi, N., Y. Song, and H. Han, A Survey on Ontology Mapping. 2006, ACM: SIGMOD.
29. Xu, T. and T. Xu, An Approach for Ontology Mapping Based on Semantic and Structural Information in Ontologies, in 3rd International Conference on Information Management, Innovation Management and Industrial Engineering. 2010, IEEE: China.
30. El-Ghamrawy, S.M., A.I. EI-Desouky, and M. Sherief, Dynamic Ontology Mapping for Communication in Distributed Multi-Agent Intelligent System. 2009, IEEE.
31. M.Ibrahim, N., M.F. Hassan, and Z. Balfagih, Agent-based MOM for cross-platform communication WorkFlow Management in SOA Systems, in ICSECS2011. 2011, Springer: Pahang, Malaysia.
32. Balfagih, Z. and M.F.B. Hassan, Agent based Monitoring Framework for SOA Applications Quality, in ITSIM10. 2010, IEEE: Kuala Lumpur.
33. Wu, Y., et al., Ontology and database mapping enables interoperability of asynchronous systems, in 2nd International Conference on Signal Processing Systems (ICSPS). 2010 IEEE: China.
34. Shi, Y., et al., Framework of Semantic Web Service Discovery Based on Ontology Mapping, in International Conference on Research Challenges in Computer Science. 2009 IEEE: China.
35. Mykka¨nen, J.A. and M.P. Tuomainen, An evaluation and selection framework for interoperability standards. Information and Software Technology, 2008. 50.