Journal of AI and Data Mining
Vol 3, No 1, 2015, 59-76.
doi:10.5829/idosi.JAIDM.2015.03.01.07

Method integration: An approach to develop agent oriented methodologies

E.Ghandehari[1], F.Saadatjoo[1*] and M. A. Z. Chahooki[2]

1. Computer Engineering Department, Science and Art University, Yazd, Yazd, Iran
2. Electrical & Computer Engineering Department, Yazd University, Yazd, Yazd, Iran

Received 4 July 2014; Accepted 7 May 2015
**Corresponding saadatjou@sau.ac.ir (F. Saadatjou).*

Abstract

Agent oriented software engineering (AOSE) is an emerging field in computer science and proposes some systematic ideas for multi agent systems analysis, implementation and maintenance. Despite the various methodologies introduced in the agent-oriented software engineering, the main challenges are defects in different aspects of methodologies. According to the defects resulted from weaknesses in agent oriented methodologies in different aspects, a combinatory solution named ARA using, ASPECS, ROADMAP and AOR has been proposed. The three methodologies were analyzed in a comprehensive analytical framework according to concepts and Perceptions, modeling language, process and pragmatism. According to time and resource limitations, sample methodologies for evaluation and in titration were selected. This selection was based on the use of methodologies' and their combination ability. The evaluation show that, the ROADMAP methodology supports stages of agent-oriented systems' analysis and the design stage is not complete because it doesn't model all semi agents. On the other hand, since AOR and ASPECS methodologies support the design stage and inter agent interactions, a mixed methodology has been proposed and is a combination of analysis stage of ROADMAP methodology and design stage of AOR and ASPECS methodologies. Furthermore, to increase the performance of proposed methodology of actor models, service model, capability and programming were also added to this proposed methodology. To describe its difference phases, it was used in a case study too. Results of this project can pave the way to introduce future agent-oriented methodologies.

Keywords: *Agent-oriented Software Engineering, Agent-based System, ASPECS, ROADMAP, AOR.*

1. Introduction

Agent-oriented software engineering is a type of engineering with agents as its main abstraction. In other words, agents are the main components of such software. The agent-oriented approach toward software engineering means dividing the problem into several autonomous and interacting agents which interact with each other to achieve the goal they have been designed for [1].

AOSE was developed to respond to the essential needs of software engineering and agent-based computations [2]. Its main goal is to create the methodologies, tools and facilities required for the simple preparation and maintenance of agent-oriented software [3]. As object-oriented software engineering (OOSE) was not able to respond to the needs of agent-oriented software, the emergent need for a new engineering compatible with agent perspectives led to the development of AOSE from OOSE [4]. One of the main challenges ahead of AOSE is that it lacks a complete software development methodology. Although a large number of agent-oriented methodologies have already been proposed, a few of them fully cover software engineering activities and none of them fully supports the development needs of agent-based systems. Therefore, it currently seems necessary to work on developing an integrated and comprehensive methodology [5-8]. In the following Paragraphs studies aimed at developing agent-based methodologies were examined.

Zambonelli et al. added the internet implementable systems modeling to the GAIA methodology. In this study, according to the openness and goals conflict in agents, the ability to model the inter agent relations was added to the model [9]. Jaunet et al. also in 2002 added a hierarchical structure for roles and developing a formal model for the system environment and developing the ability to manage dynamic changes to GAIA methodology and proposed a new one named ROADMAP [8]. Another development in this methodology was performed by Garcia et al. enhanced the interaction, agent and protocol stages and the UML developed model by a combination of this methodology and the AUML [10]. Gonzalez et al. also tried to enhance this method logy by adding agent design stage and a repeatable approach [11]. Agent oriented methodologies enhancement is not limited to GAIA and is continuing on Methodologies like MASE and TROPOS yet. In one of MASE enhancements the ontology stage was added to the analysis stage of the methodology by Dieloet et al. [5]. Another extension named organizational relations modeling was added to the MASE methodology [12]. In TROPOS methodology, a formal goal analysis model was added [13]. A method to associate goals with roles was also added to this methodology [14].

According to these deficits, we first tried to extract some positive and negative properties of ASPECS [15], ROADMAP [8] and AOR [16] methodologies and then a combinational methodology using these properties were proposed. We also tried to achieve a good convergence by segregating models in different analysis and design phases and propose some models (agent, capability and programmer model). This convergence enhances the proposed methodology's abilities and paves the way toward future generation of agent oriented methodologies. To do so, we should continue our research to develop and enhance agent oriented methodologies.

In order to segregate object oriented and agent methodologies we will examine the differences between objects and agents. Then in section three, we pointed out some evaluation indices. In section four, we will introduce selected method in this article. In section five we identify the proposed methodology and analyze its different phases and finally in section six. We propose conclusion and future works in this area.

2. Comparison between object-oriented and agent-oriented approaches

AOSE has evolved from OOSE. In other words, agents have been derived from objects [17]. LIND compared object-oriented systems with agent-oriented ones in terms of hardware, theory, implementation time, programming language, and designing language [18], producing the following results: a)Objects have a central structure but agents perform distributed computations. b) Objects are more homogenous than agents in a system. c) Agents cannot initiate or destroyed as easily as objects. d) the object's behavior and structure doesn’t change but agents learn from their experiments and change their behavior e) objects' interactions are usually a result of the other object demand but agents have their own reactions in front of the environmental or other motivators or the other agents' demand f) objects' interactions are usually synchronous but in agents it is vice versa. g) Encapsulation in agents is stronger than objects.

Since agents are derived from objects, there are also similarities between them. Parameter from both approaches though these similarities and differences could be mapped together. Table 1 presents a typical mapping of object-oriented and agent-oriented approaches.

Table1. Mapping of object-oriented and agent oriented approaches [18].

Object-Oriented Approach	Agent-Oriented Approach
Abstract Class	Generic Role
Class	Domain-Specific Role
Class Variables	Knowledge, Belief
Method	Capability
Inheritance	Role Binding
Prototyping	Specific Role + Personal Knowledge
Compound	Holon Agents
Method Invocation	Message Exchange
Cooperation	Interaction

Table 1 shows that the agent-oriented method has a solution for all object oriented methodology abilities. These solutions are suitable for agent - oriented systems analysis and design.

3. Criteria and evaluation methods

The first and the most important step in every analysis is) to determine its goals [19]. In this study, two purposes for evaluations were proposed. The first objective is to determine the strengths and weaknesses and similarities and differences of the studied methodology to enhance a developed agent-oriented software system. The second objective is to equalize agent oriented methodologies by a combination of their strengths and also proposition of a way to enhance their limitations.

Then, the evaluation framework, which is built, which consists of properties, attributes and measures. Measures used here to evaluate OOSE methodologies are based on current works and other works performed on agent-oriented methodologies [19-22]. The evaluation was focused on the technical properties of the methodologies. Determination of models and common projects were compared with three methodologies and their significant aspects were evaluated. This subject plays an important role in the next generation of agent-oriented methodologies development.

Next, a methodology evaluation framework is proposed according to properties comparison. It consists of some measures and rules that cover AOSE exclusive features too. Figure 1 shows the agent-oriented methodologies evaluation framework. Figure 1 illustrates the 4 aspects of the evaluation named concepts, modeling language, processes and operation orientation.

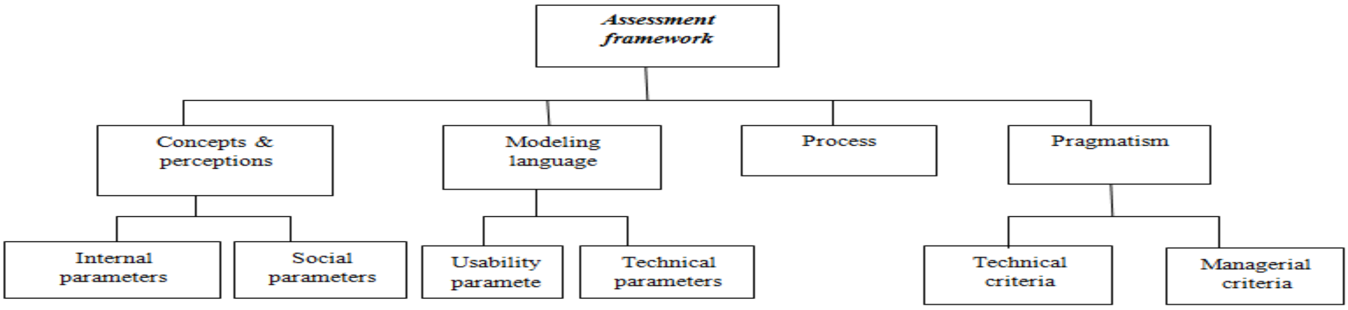

Figure1. General framework of agent oriented methodologies assessment.

Each proposed index is divided into some measures. These indices are discussed comprehensively in table 2 with full details and measures. To evaluate the proposed methodologies according to these 4 indexes, we proposed some questionnaires to experts and analyzed the corresponding answers according to fuzzy Delphi method [23]. We will discuss the results of these evaluations and the answers of the questionnaires next.

3.1. Validity, reliability and measure of questionnaire

To determine the study justifiability, we first performed the basic test for the questionnaire. So, the initial questionnaire was distributed between 11 expert teachers in agent-oriented software engineering field. After collecting the answers, some obscure and unrelated questions were determined, edited, reformed and placed in the final questionnaire and those questions totally unrelated were removed. To determine the questionnaire sustainability, we used the cronbach alpha coefficient. The calculated value for this variable in this study was 0.832 that is acceptable according to the research principles. The measurement scale in this study was the Likert 5 Points measure.

Methodologies investigated by provided parameters in evaluation framework, were analyzed by fuzzy Delphi method. This section investigates similarities and differences of these three methodologies. So similarities areas of these methodologies are Goal models Co-execution, plans, static & dynamic structure and record model. Table 3 will show similarities of 3 models.

Region of differences among three methods according to the assessment of results includes basic needs and the environmental model in ROADMAP methodology and sustain model of the methodology in ASPECS. Table 4 presents the differences in methodology.

Then, according to the results of the assessment, we will have a case study and introduce ARA combined methodology. In order to assess the ability of the proposed methodology, a case study will be used.

3.2. Findings and evaluation results

In the previous section, methods and frames used to assess was described. The principles used in the assessment, indicators were defined in four parts. In this section, using the criteria of evaluation results is presented. The results of the analysis are based on fuzzy Delphi method. Due to the nature of some of the benchmarks, only the presence or absence of the methodology is reviewed.

- The first indicator: Concepts and perceptions

Property type 1: internal properties

1- Autonomy: autonomy is a key feature for agents. It differentiates them from other entities. According to evaluations, all the three methodologies here have this feature.

Table 2. Concepts and properties covered with evaluation framework indexes.

Index definition of the concepts and features	Index definition	Index
□ Internal features Autonomy, mental attitude, goal oriented, response, the Co-executive, located in the environment. (The methodology supports the mechanism of self-control, and a range of features, models, objectives, changes in the environment, parallel processing and features an internal model of the environment is checked) □ Social features Means of cooperation, teamwork, protocols, communication languages (The methodology supports the model of cooperation, teamwork agents display methods and forms of expression and communication protocol between agents of social features to be checked).	Agents and agent oriented systems concepts	Concepts and Perceptions
□Usability features Intelligibility and clarity, distinctness, ease of use □ Technical features Compatibility, the ability to track, refining, reusable (The methodology supports the modeling techniques to examine the compatibility of the path analysis modeling activities to implement, develop processes and mechanisms for reusing existing components in the form of technical features to be checked). Software life cycle stages and activities □ cover the full life cycle Including the development of methods designed to duplicate the top-down and bottom-up grant.	Modeling language for model illustration	Modeling language
□ coating process with sufficient detail Including definitions, initiatives, decision management, quality assurance guidelines and estimates □ Support the development of concepts Includes reusable and Prototyping	Software life cycle	Process
□ management features Cost (Property management fee examined the methodology adopted). □ Technical features Applicable range of scalable, flexible distribution (The support of the use of different methodologies, different size and design of systems-management and distribution of technical specifications will be reviewed).	Methodology development aspects	Pragmatism

Table3. Methodologies similarities structure analysis.

Model goal	Model covering methodology	Comparative model
Goal model	ASPECS ، ROADMAP ،AOR	Goal achievement
Co execution model	ASPECS ، ROADMAP	Agent modeling independently
Role model	ASPECS ، ROADMAP ،AOR	Agent role definition
Static and dynamic model	ASPECS ، ROADMAP	Agent internal architecture design
Pre model	ROADMAP ،AOR	Agent relationship model

Table 4. Analysis of differences between methodologies.

Model / Difference model	Model covering methodology	Model objective
Basic needs/ environment model	ROADMAP	requirements /domain demand
Allocation model	ASPECS	Structure definition and relationships

The support level for this parameter is acceptable in all of them. All of them support this property and have some functions and enhancements in agents for that. In addition, the co-execution plot in ASPECS and ROADMAP makes it possible to model agents free from their environment and other entities.

2- Mental orientations: ROADMAP methodology supports these parameters fully with its internal functions and illustrates the agents' knowledge from their environment. The agents' goals are also modeled in this way. On the other hand, ASPECS and AOR have a weaker support for this parameter. ASPECS has the goal plot but cannot illustrate the agents' knowledge.

3- Goal orientation and intractability: The evaluation of the measurements results for these two parameters are difficult. It is well supported by some methodologies. Like previous parameter, these two methodologies get the goals and then perform some operations to achieve them.

4- Co-execution: Support from this parameter varies between methodologies. It varies from bad to good. From experts' point of view, ASPECS has the best support for this model. In this methodology, a single role can be co-executed.

5- Be in environment: Support for this parameter varies from average to good among different methodologies. Experts believe that ROADMAP outperforms others in this respect. They believe that AOR is the worst in this respect. They believe that this is because the AOR doesn't support the environment model.

Property type 2: Social properties

1- Cooperation and team work methods: for this parameter we evaluated the multi agent programming and the team work. In ROADMAP and ASPECS the creators argued [8][15] that these methodologies support general agent oriented cooperation and any other kind of cooperation can be driven from them. But, in experts' point of view, none of these parameters are covered with these methodologies explicitly.

2- Protocol: ASPECS methodology with its analyzer protocol outperforms the two other methodologies. AOR doesn't provide a special model to show protocols, but shows interactions among agents in high levels. ROADMAP doesn't have any explicit definition for protocol except in AUML [24].

3- Communicational Language: Experts believe that, this feature is in all three methodologies. Since the interaction among the agents has some levels of knowledge. (All three agent-oriented methodologies as the communicational language have the aim of speech act).

- The second index: Modeling language

Feature type 1: Usability features

1- Intelligibility and clarity: The measure of how brightly the symbol definition specifies the syntax and symbols models. Symbol provided by the three methods are well understood.

2- Distinctness: The number of static and dynamic models and the different views that show the destination system are good test for these measures. ASPECS methodology of aspects of system dynamics model and protocols, deal with ROADMAP methodology for system dynamics modeling protocols with the exception of some support in the detailed design level, which does not provide strong support. However symbols in the ASPECS methodology look meaningful. However, this methodology does not provide different views of the destination. AOR methodology has models for static and dynamic aspects of the target system and sees the system from a different angle. Experts believe that, modeling language of AOR methodology is not suitable because it does not give you the detailed structure. AOR-oriented methodology is not actually a perspective- oriented methodology.

3- Easy to use: According to experts, and connoisseurs', opinion, all three methodologies are a symbol and using them is simple.

Feature type 2: Technical features

1- Compatibility: for controlling the terms of compatibility, the methodology is tested at different levels. ASPECS methodology supports it well; While ROADMAP and AOR methodologies don't support it. From the perspective of qualified professionals, reason of this weak support is accessibility to supportive tools.

2- Ability to track: Similar to compatibility criteria, ASPECS methodology supports this feature. This methodology provides a clear link between their models. For example, goals, roles, and operating practices will bond together. These connections allow the developer to obtain a model of the design (e.g. interior architecture of agents).

3- Refinement: three methodologies don't have a proper support from this standard. From the perspective of professionals and experts, this issue reflects the fact that the language of modeling three methodologies isn't integrated. In fact, the developers can move through the phases and add details to the model.

4- Reusability: None of the three tested methodologies use techniques explicitly to support the design and use of reusable components. Also, the reuse of existing components in each methodology is not seen.

- The third indicator: The process

1- Development Principles: when looking for the life cycle of software development, it is clear that all three studied methods have architectural design and detailed design. Except for AOR methodology, implementation is supported in two other methodologies. Test and debug are only special for ASPECS methodology. ASPECS is the only methodology which describes the development of the operating system and it is part of the design phase. In view of the development, the methodology ASPECS supports top-down and bottom-up method. While ROADMAP methodology and AOR are appropriate for top-down approaches.

2- The process stages: These processes are described well in analysis and design phase of methodology ASPECS and ROADMAP. While descriptive design in AOR methodology isn't documented well. So from views of experts, this issue is resources' constraints in the AOR methodology.

3- Developmental Support concept: There are several key concepts such as prototyping and reuse of components there. From experts' point of view, none of these three methodologies have subject related to sampling in process or creating a reusable component.

4- Quality assurance guidelines and estimates: Due to lack of agent-oriented methodologies development and from the experts' perspective, a detailed statement can't be done for this parameter.

- The fourth indicator: Pragmatism

Feature type 1: managerial features

Cost: The cost of achieving methodologies and supportive tools is free for all methodologies and its documentations are accessible.

Feature type 2: Technical features

1- Domain applicability: From experts' point of view, there is no limitation for domain of applications of these three methodologies. These domains for an agent-oriented system with autonomic software are reliable and powerful.

2- Scalability: None of these methodologies cover this parameter. From experts view, none of them has proposed anything about this aspect and how it is defined.

3- Distribution: From experts' point of view, ROADMAP and AOR support this parameter implicitly. In their opinion, ASPECS is an exception and the design stage of this methodology, makes it possible to design and allocate agents in the network. That is because of the allocation model.

4. Case study: Housing sales system

In this article, the system of buying and selling house is defined in an online frame and will provide different sections necessary for buying and selling. In this system, people can see information about done trades by searching on site and then decide about buying or selling house. When buying, you should have one third of the amount specified by the Department of Housing and Urban Development and when giving selling request, providing valid document to experts is required (all the stages are done electronically). Buying and selling requirements involve determining what the size, location, year of construction are and how to do deal in terms of with the price; it means that the maximum or the minimum purchase price (buyers and sellers) will be defined. Buyers and sellers can also put provisions in their requirements. For example, according to variable price of house market, seller can choose a special month to show house information in order to have more profit (since in some months house has better price) and this can be a strategy from seller. On the other hand, if the buyer wants to pay money in installments, he can say it. After buyers and sellers requirements, a department's expert will check the trueness of their requirements and the information will be recorded in system if they are true and precise, and after entering the requirements in system, the system will organize housing transactions according to priorities, and according to the restrictions imposed by the parties, buyers and vendors can provide a list of the items in the next 48 hours (minimum and maximum price and other conditions) according to which a deal can be done. After viewing the list, buyers and sellers can choose and cases on the base of priorities. In requirement, the priority of a deal is on first requirements (both among buyers and sellers). After doing the deal, a housing system will issue deal documents which are certification of department's expert, buying certification, selling certification and temporary certification of deal. After preparing the documents, the relevant certifications will be delivered to the parties. In this system, fees for each transaction will be divided between the parties based on the Ministry of Housing and Urban Development Act.

5. Introduction the methodology of ARA

ARA is an agent-oriented software development methodology that is made of a combination of three ASPECS, ROADMAP and AOR methodologies and actor, planning and capability proposed models. In determining the phase of ARA methodology, three ideas were used are: a) perception and understanding of the agents and mental imagery (objectives and program-planning) during the analysis phase of software development, from basic analysis to used design. b) in order to have a complete understanding from system and environment, the actor model is added to phase analysis of ARA methodology.c) to clarify agents' capabilities in detail and to deter the way of performing these capabilities by agents, two capability and programmer models are proposed in ARA methodology.

Analysis and design of actor consists of a big set of concepts, so understanding all aspects of analysis and design model from a special view is difficult. For this reason, in an ARA methodology, several models that focus on various aspects are defined. These models have various aspects, but are not complete alone, so by putting them together, a complete and understandable view of the system will be achieved.

In choosing models of ARA methodology, noted earlier, a complete evaluation frame is used and covers four main areas of agent-oriented software engineering such as [19-22] concepts, modeling language, process and activism. According to this frame and its parameters, the support of each methodology is evaluated and experts investigated the actor, capacity and programmer proposed models. Thus models used in the ARA methodology of agent-oriented concepts are largely covered and these models do not overlap with each other. Table 5 shows the differences between an ARA methodology and the three invested methodologies.

Table 5. Structural analysis of the difference between the ARA methodology and studied method.

Models/ Difference models	Model goal
Actor model	Fully understanding the environment
Capability model	Determine the Agent capabilities
Programming model	Implication of capabilities by each agent

In order to assess the ability of the proposed methodology, it is used in a case study that is described in Section IV.

5.1. Housing sales system analysis using phases of the proposed methodology

In this part, a method is provided to homogenize three methodologies of ASPECS, ROADMAP and AOR by combining the strengths and avoiding its limitation. In fact, to create a new methodology, some parts of this methodology are used according to the framework presented in the previous step. Using combined method can affect the proposed methodology in order to covers most of the Agent-oriented software engineering and it will be effective in developing next Agent-oriented Methodologies This article used developmental process to combine methods and according to importance of analysis and designing phases in creating qualified and reliable software products, it will focus on these two phase.

5.1.1 Phases of the proposed methodology

1. Analysis phase

The proposed methodology in analysis phase consists of actor model, goal model, knowledge model, environment model and role model and these five models will provide a strong support for defining relationship among system actors, targets, duty and knowledge of system, obtaining environment and defining key roles in system. These models will increase knowledge of developers about system requirements and will provide inputs for next steps (The analysis and design of a system is done by using AUML diagrams).

-Actor model

Organizing and defining the actors of the system under study is one of the important steps in the analysis phase. In this model, the physical attributes of the system are identified. Figure 2 shows the actor system of buying and selling real estate. As you see in figure 2, buying and selling system of house has three actors of seller and buyer / department expert and the Department of Housing and Urban Development.

Each of identified actors in the system has goals.

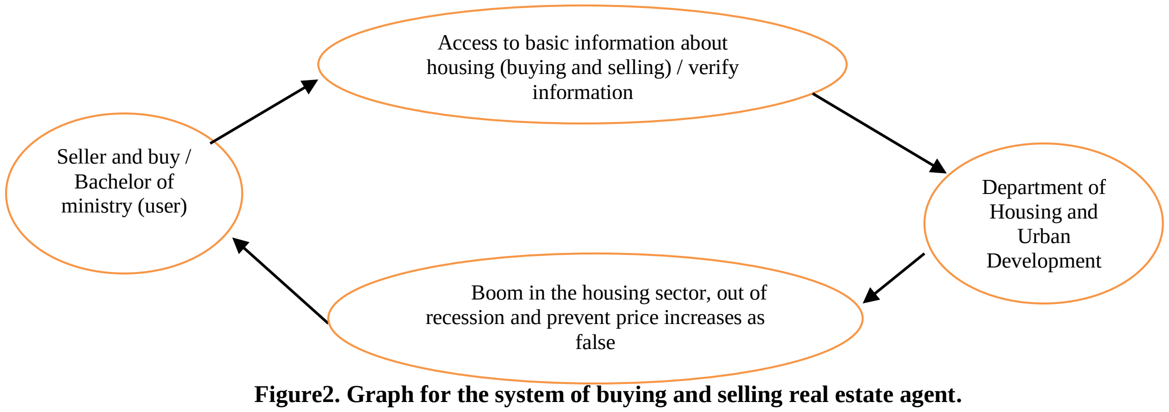
Access to basic information about housing (buying and selling) / verify information
Department of Housing and Urban Development
Boom in the housing sector, out of recession and prevent price increases as false
Seller and buy / Bachelor of ministry (user)

Figure2. Graph for the system of buying and selling real estate agent.

The aim of the seller and buyer agent / department specialist, is access to basic information about housing (purchase and sale) and / the actor objective of expert is to investigate the trueness of buyers and sellers' information. The actor purpose of the Ministry housing boom, is to go out of recession and prevents a rise in housing prices. Using the graph of the relationship between the active agents can be clearly displayed by the system.

- Goal model

Organizing and identifying goals, are an important step in extraction of requirements. In this model, the overall objectives of the system are determined in different levels.

Figure 3 shows the purpose model and functions of the system of buying and selling property.

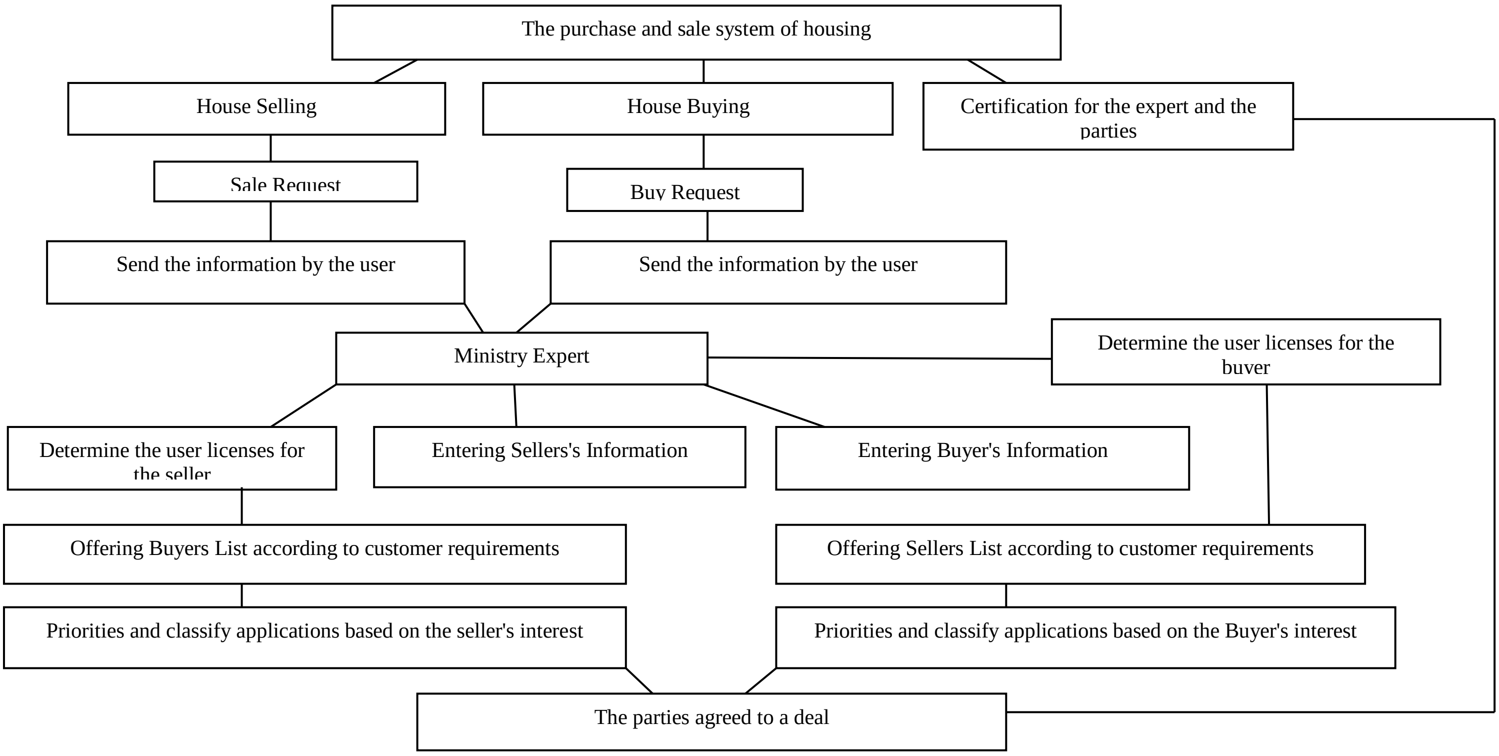
The purchase and sale system of housing
House Selling
House Buying
Certification for the expert and the parties
Sale Request
Buy Request
Send the information by the user
Send the information by the user
Ministry Expert
Determine the user licenses for the buver
Determine the user licenses for the seller
Entering Sellers's Information
Entering Buyer's Information
Offering Buyers List according to customer requirements
Offering Sellers List according to customer requirements
Priorities and classify applications based on the seller's interest
Priorities and classify applications based on the Buyer's interest
The parties agreed to a deal

Figure3. Goal model and the purchase and sale system of housing.

As figure 3 shows, house selling and buying system include three parts of buying home, selling home and issuing a license. In this system each of buyers and sellers should first send requirements in order to buy or sell a house. After the requirement is met, all information should be sent completely and truly and after that the expert will confirm the trueness of this information. If there is no problem with information, user license will be issued and related information will be put on it. After this stage, a list will be given to them (list of buyers for sellers and vice versa) that they can use it by the users license and according to this list, the priorities and interests of buyers and sellers will be categorized and if they agree, system will issue the license.

- Knowledge model

A model of system has rules, procedures and limitations. Rules relate to principles that system will make decisions on their basis. Procedures will clarify the performance of system and limitations are necessary for using a system. Table 6 shows the knowledge model for the system of buying and selling a house.

Table 6. Knowledge model of the buying and selling housing system.

	-If the purchase price is equal to or greater than the price the seller, relevant housing information will be displayed to the buyer
Rules	-Provide evidence of Salable Property by seller
	-If the seller price is equal to or greater than the price the buyer, information relevant to buyer will be displayed to the seller
Steps	-The purchase of housing step: Having a third floor price set by the Department of Housing and Urban Development, Offering buy request, taking User license for listed and classified according to priority
	-Offering bank account by the buyer met the third floor of an amount determined by the Department of Housing and Urban Development
Limitations	-Offering valid proof of ownership by the seller
	-Intervals to provide a list of buyers and sellers, are 48 hours after entering information by the ministry's expert

- Environment model

An environment Modeling is used to clarify the scope of the system and functions are specified the recognition of systems. In that respect, the environmental agents and their relationship with each other will be determined by the system. Also, in this stage the components of system. Show the environment model in the system of buying and selling house (see Figure 4).

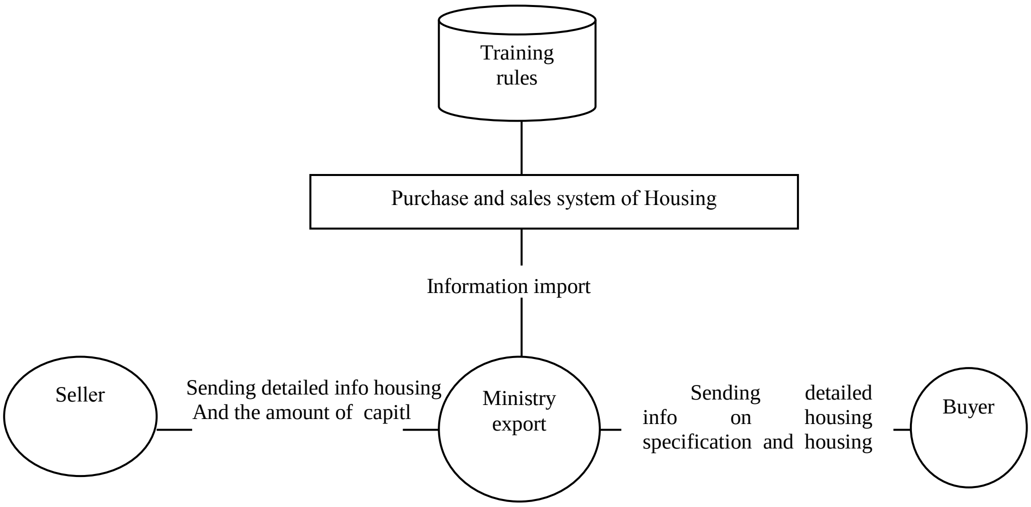
Training rules
Purchase and sales system of Housing
Information import
Seller
Sending detailed info housing And the amount of capitl
Ministry export
Sending detailed info on housing specification and housing
Buyer

Figure 4. Model for the environment of purchase and sale of housing.

- Role model

In the agent-oriented approach, the agent is considered as a key entity. Thus it can be said that one of the basic needs of the agent-oriented methodology is helping developers to specify the agents of system. In the proposed methodology, a role model technique is used to specify agents, so the roles in systems are extracted precisely and then according to specified roles they are defined. In fact, the role model is the main part to determine the agent, because the agents should be used in the system to carry out their roles. This model includes the objectives, the sub-roles and responsibilities of the role in the system. The system of buying and selling house includes search, storage and retrieval of information, and user licenses, classifying information and updating information will be provided in depth.

- **Role of searching information**

This role is defined to enable users to have a search about lists. Thus users can search detailed information by clarifying basic information (national code). Since users have to define some details about the house like its size, year of its construction, and other similar things, so buyers and sellers can see needed information by searching the national code of person from the list. Also users can search their own or other information about priorities. Table 7 shows the role model for searching information.

Table 7. The role model for the searching information.

Role name: Search of information

The Purpose of the role: Providing information on the national code entered by the user

Details of role: Read limitation information (entered national code) that Search operations must be performed based it

Responsibilities: Offering list based on respective national code

- **Role of storing and recovery of information**

This role is in the frame of storing information according to priorities of users' interests, storing background information that includes all operations that user has done on system till now and storing user information. Table 8 shows role model for storing and recovery information.

Table 8. Role model for the storage and retrieval of information.

Role name: Information storage and retrieval

The Purpose of the role: Store information of preferences, history and user

Details of role: Read the information entered by the user

Responsibilities: Create a list of information for user

- **Role of user license**

This role is for experts, buyers and sellers that want to enter the system. In fact, the role of user license is to confirm the user information. Table 9 shows the role model for user license.

Table 9. Role model for the role of use licenses.

Role name: User License

The Purpose of the role: User authentication, create the permissions

Details of role: Read the information entered by the user and query the database to test of the validity

Responsibilities: Protection of system Security

- **Information classification**

Information Classification is defined based on priorities of user's interests. This role has two lists of level 1 and level 2 that user interest are in level1 and other cases are in level 2. Table 10 shows role model for classification role.

Table 10. Role model for the classification role.

Role name: Information Classification

The Purpose of the role: Information Classification based on priorities of user interest in both one and two level

Details of role: Read user preferences

Responsibilities: Create classification in user Priorities at two levels one and two

- **Role of updating information**

This role provides a list of new added materials to sell and buy house and deals with information and updating list of users' priorities and interests. This role will fetch selling and buying information from database of buying system and fetch desired information from database of urbanization and housing department and provides it for users in some lists. Table 11 shows role model for updating information.

Table 11. Role model for the update information role.

Role name: Update information

The Purpose of the role: Update information and display the status of transactions, update interests and priorities

Details of role: Reading the information from database of buying and selling system of housing and ministry of Housing and Urban Development and User Interests

Responsibilities: Rendering the list of housing added for buying and selling and transactions, Changes in Wish list

In analysis stage, according to target and duty, knowledge, role and environment models, system requirements and its rules and limitations were extracted.

2. Design phase

The proposed methodology in designing a phase includes: agent model, interaction model, capacity model, programmer model and service model. In the agent model, roles are played in any agent. An interaction model will clarify relations between agents to do play roles. Capacity and programmer models are used to modeling the capabilities of agents and to program the way of making these capabilities. The service model point out capabilities that each role should have.

- Agent model

This model is the supplement of the role model in the analysis phase. In this model according to defined roles in role model, an agent will be defined and then each role will be written in an agent. Buying a house and selling system consists of five agents that in table 12, the roles are played written in this agent.

Table 12. Allocation of roles to agent.

Search operation	→	Search agent
Save and restore operation	→	Save and restore agent
Operation of User License	→	User License
Classification of the user's interest operation	→	Classification agent
Update information Operation	→	Update agent

- Interaction model

This model clarifies relations between agents to play roles. In other words, an interaction model will model the way of doing playing roles. In role model 5 roles will be defined for system. The first role is information searching whose interaction model is shown in figure 5.

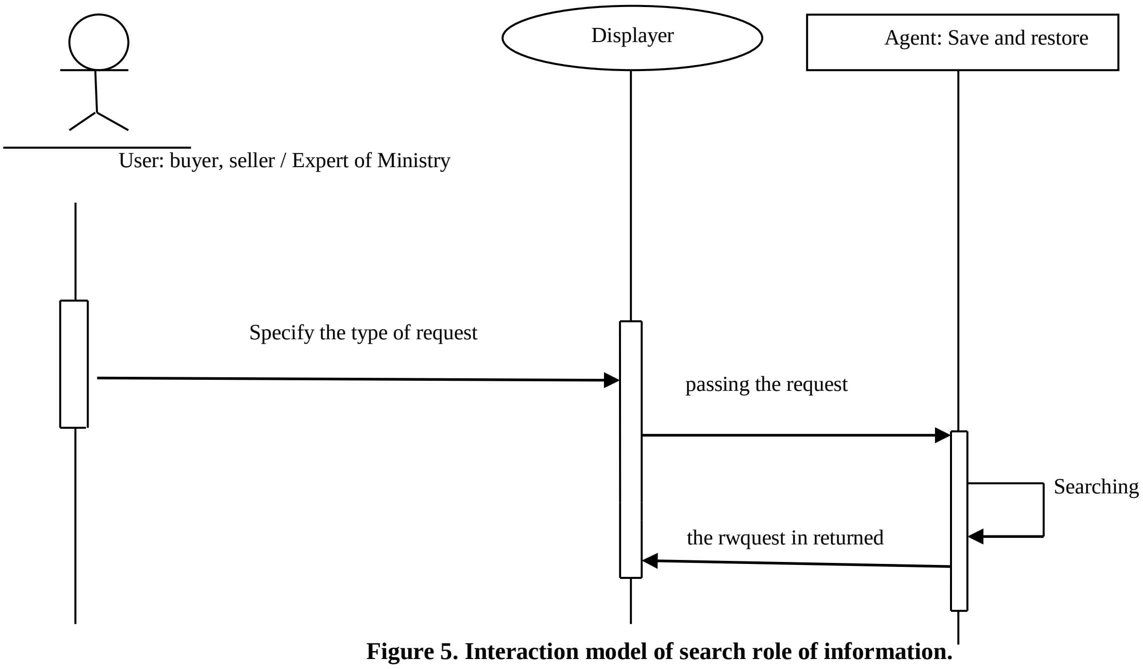

Figure 5. Interaction model of search role of information.

According to figure 5, a user (buyer/seller/expert) will first define the kind of his requirement. Then this request will be given to search agents and this agent search the kind of request from system database and show the result.

Another roles that will be clarified in the role model, is the role of storage and recovery. Figure 6 shows the interaction diagram of this role.

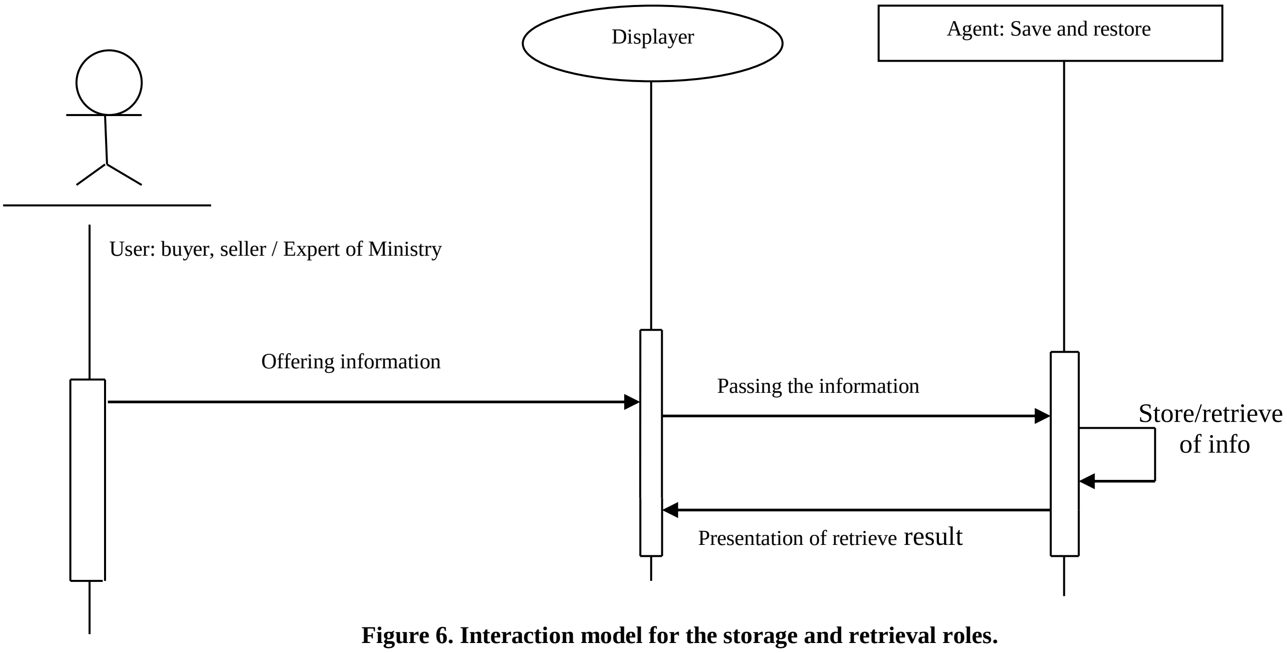

Figure 6. Interaction model for the storage and retrieval roles.

According to figure 6, a user (buyer/seller/expert) should first provide information that need to be stored and recovered, then this information will be given to storage and recovery agent and this agent will make the storage and recovery based on the information and shows the results.

Next, we will discuss interactions models of user license role. Interaction models of user license role are under investigating from two aspects of storing user information and confirm the validity of user's information.

In storing user information, a user has to enter his account information and this information will be given to a user's license agent. This agent will evaluate the information (that passing code and word is correct or not) and if the code is incorrect, an error alarm will be made and information will be back.

If there isn't any problem, information will be sent to storage and recovery agent and this agent will store the information. Figure 7 shows the interaction model for a user license role in the scope of storage.

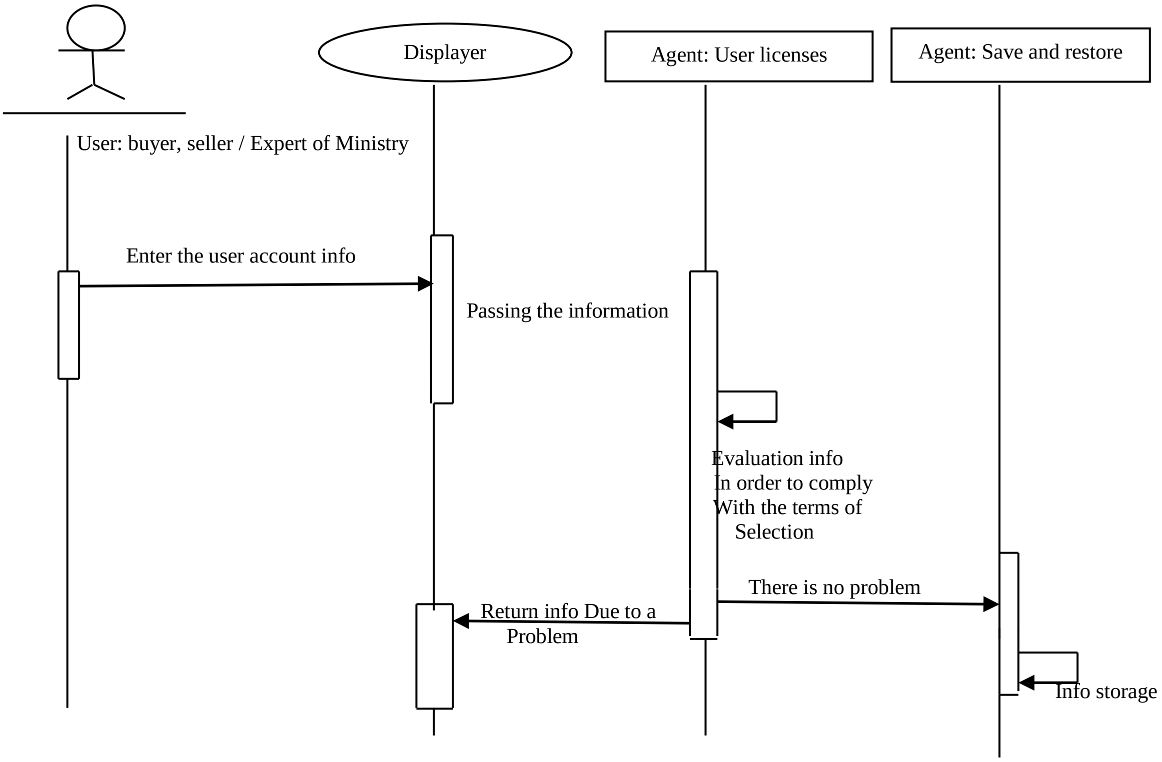

Figure 7. The interaction model of user licenses role within the scope of store.

It should be mentioned that a user (buyer or seller) should register in system just when he/she wants to send a buy/sell request. After requirement is met and if there wasn't any problem in it, experts will give a special password and code to them and last password and code are not needed any more.

For confirming a validity of user information, a user enters his user's information to system and this information will be given to a searching agent and this agent will do searches in system database. Figure 8 shows the interaction model of user license role in the scope of confirming validity.

Another that is clarified in role model is the role of classifying information. In the interaction model of this role, first of all the users will specify what is in his mind in the list of houses, then this information is given to classification agent who will classify information in level 1 (user interests) and level 2 (other cases). Figure 9 shows the interaction model of information classification.

About updating the role, a user has to clarify the kind of updating (updating information of houses, updating interests of user).
After the kind of updating is clarified, the request is given to searching agent who will search the information based on the updating kindly and results are given to update agent and this agent will update on the list and send information to users. Figure 10 shows the updating role of the interaction model.

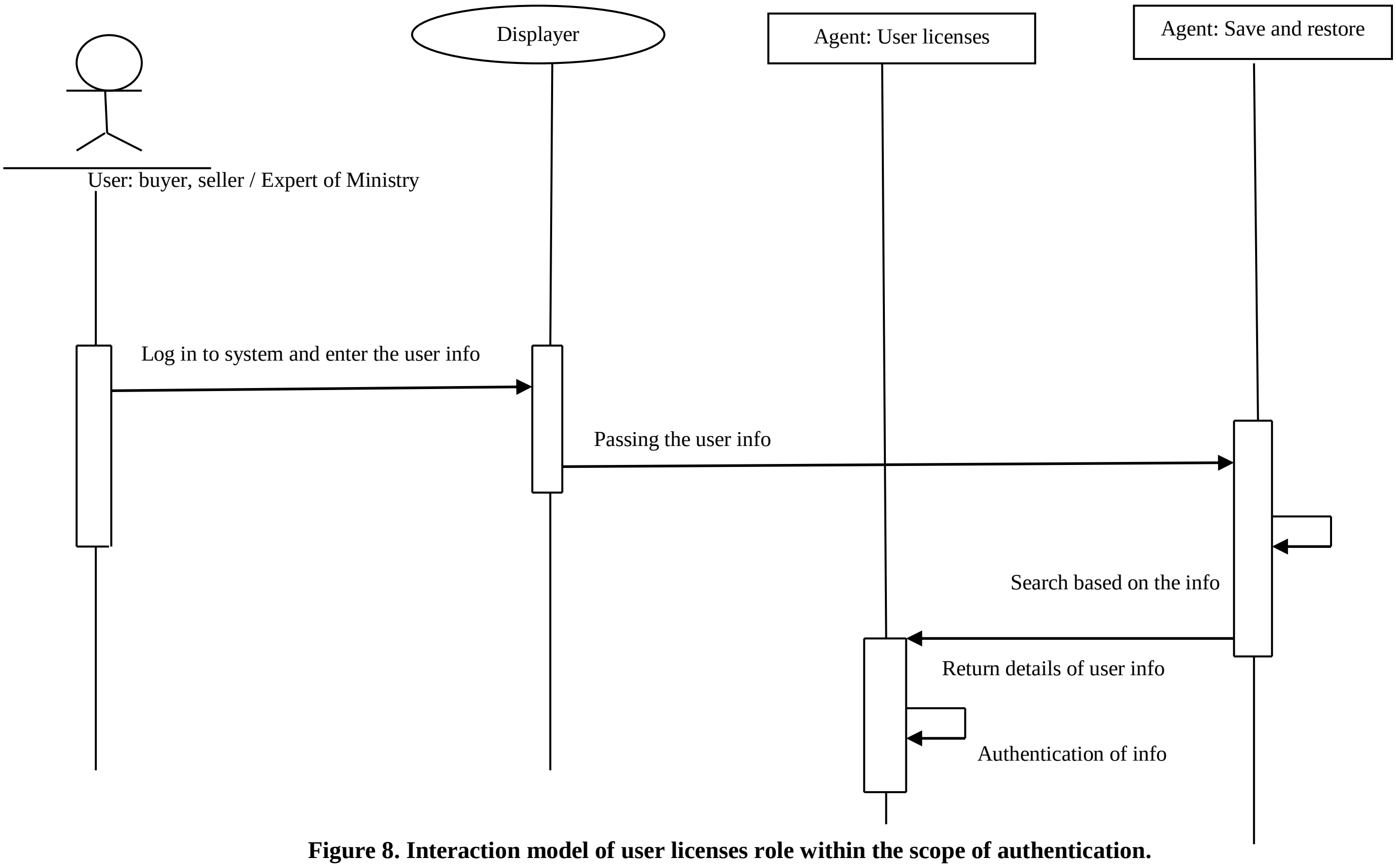

Figure 8. Interaction model of user licenses role within the scope of authentication.

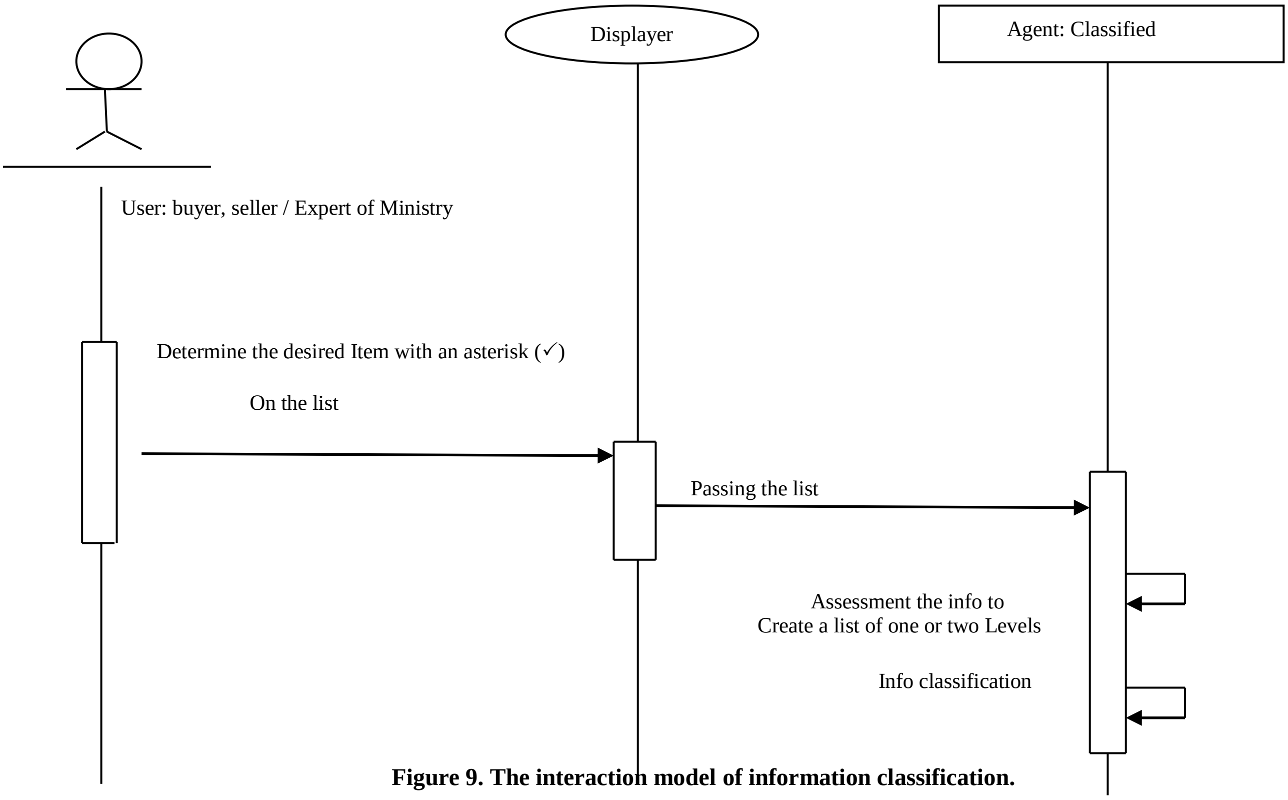
Displayer
Agent: Classified
User: buyer, seller / Expert of Ministry
Determine the desired Item with an asterisk (✓)
On the list
Passing the list
Assessment the info to
Create a list of one or two Levels
Info classification

Figure 9. The interaction model of information classification.

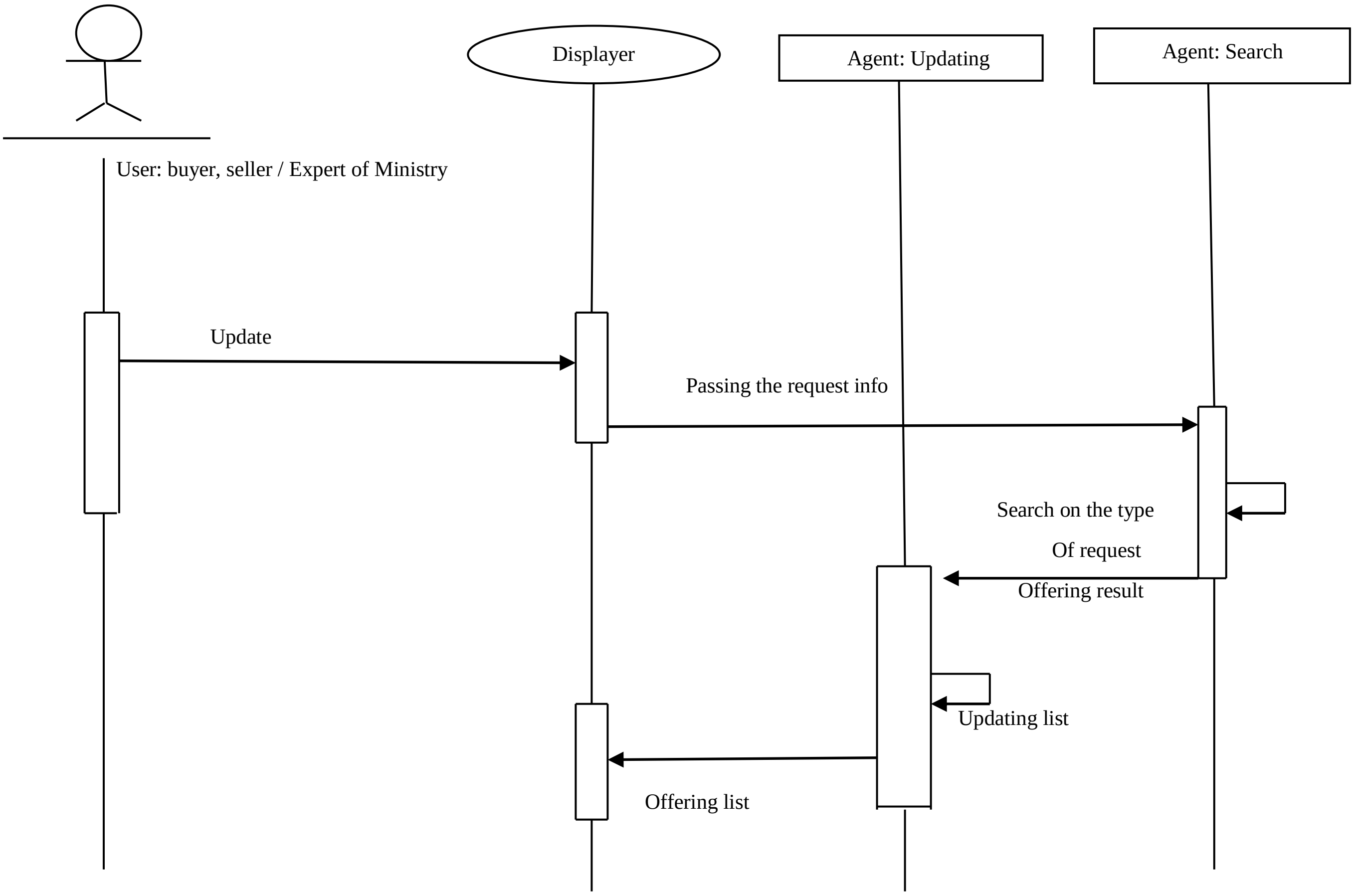

Figure 10. Interaction model on the updating role.

-Capacity and programmer model

This provides model capabilities for agents. Also models will program the stages of performing capabilities. In other words, in this model, the range of agents' duties and the way of doing them are modeled. In the agent model part, five agents were selected that first of them is information searching agent that capability model of this agent is shown in figure 11. Also the programmer model of this agent is shown in figure 12.

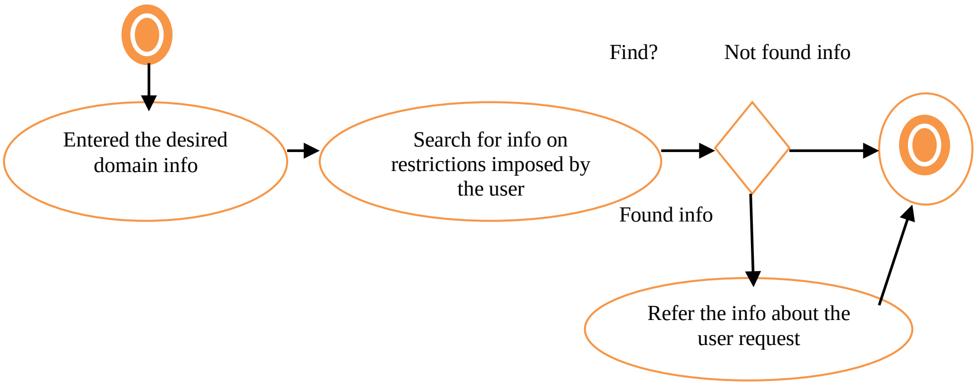
Find?
Not found info
Entered the desired domain info
Search for info on restrictions imposed by the user
Found info
Refer the info about the user request

Figure 11. Search capability model.

According to figure 11, the capability of searching agent is modeled according to information provided by users while figure 12 will model this capability by using searching agent.

Another agent recognized in the agent model was storage and recovery agent that didn't need to be modeled through capability and programmer models, since it doesn't have any exception and is modeled in the interaction model in a good way.

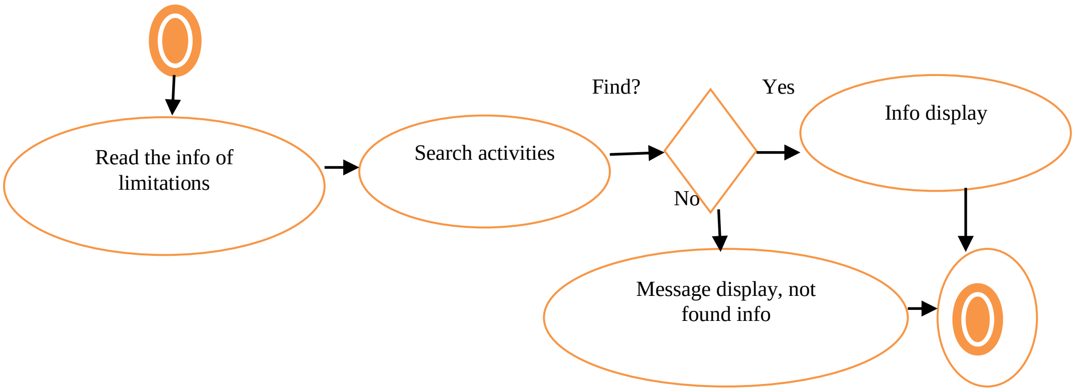

Figure 12. Planner model of search agent.

Next, we will discuss capability and programming model of user license agent. As it was said before, a user license agent plays the role of confirming validity of password and code. So if theses password and code are correct, you can access to system, otherwise, you are not permitted and an error message will be shown. Figure 13 shows the capability model of user license agent.

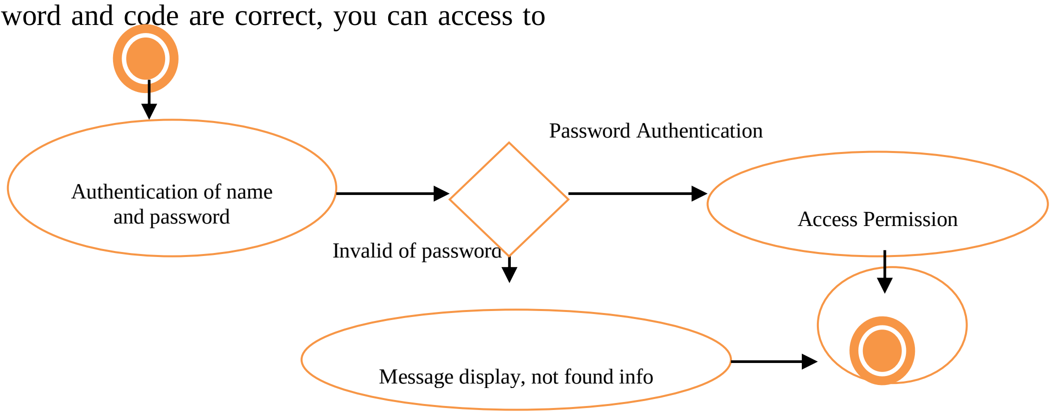
Password Authentication
Authentication of name and password
Access Permission
Invalid of password
Message display, not found info

Figure 13. Capability model of user License agent.

In programming the model, the user license agent will read password and code first and then will search for it. If the entered password is correct, entered password will be compare with database password after finding information. Figure 14 shows the programmer model of user license agent. Another agent specified in agent model was information classification agent. In capability model of this agent, first, the user will show his interests according to provided list and classification agent will classify these interests on the base of their priorities

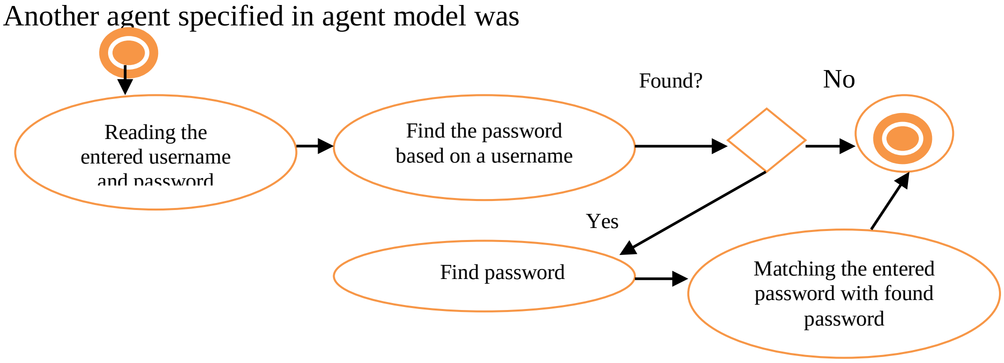
Found?
No
Reading the entered username and password
Find the password based on a username
Yes
Find password
Matching the entered password with found password

Figure 14. Planner model of user licenses agent.

Figure 15 shows capability model of classification agent. In programming the model, first, the classification agent will receive interest information of user and create two level of 1 and 2, then it will separate information according to liked and disliked priorities, as information about interests are in level 1 and information about dislikes are in level 2. Figure 16 shows the programming model of classification agent.

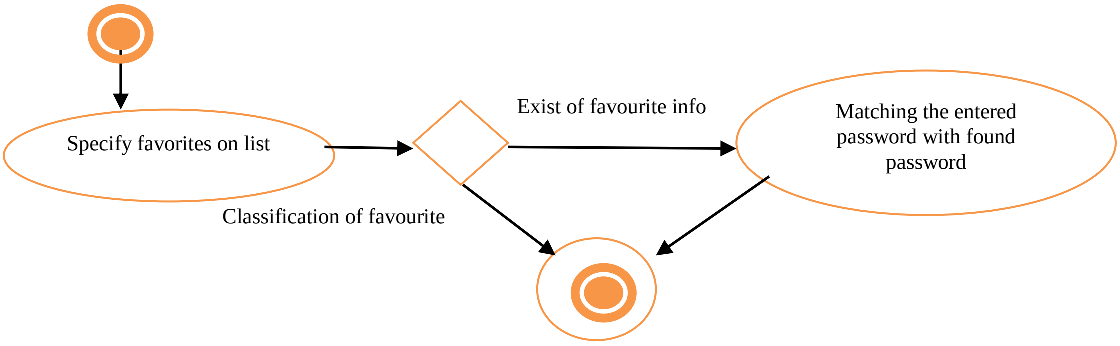
Specify favorites on list
Exist of favourite info
Matching the entered password with found password
Classification of favourite

Figure 15. Capability model of the classification agent.

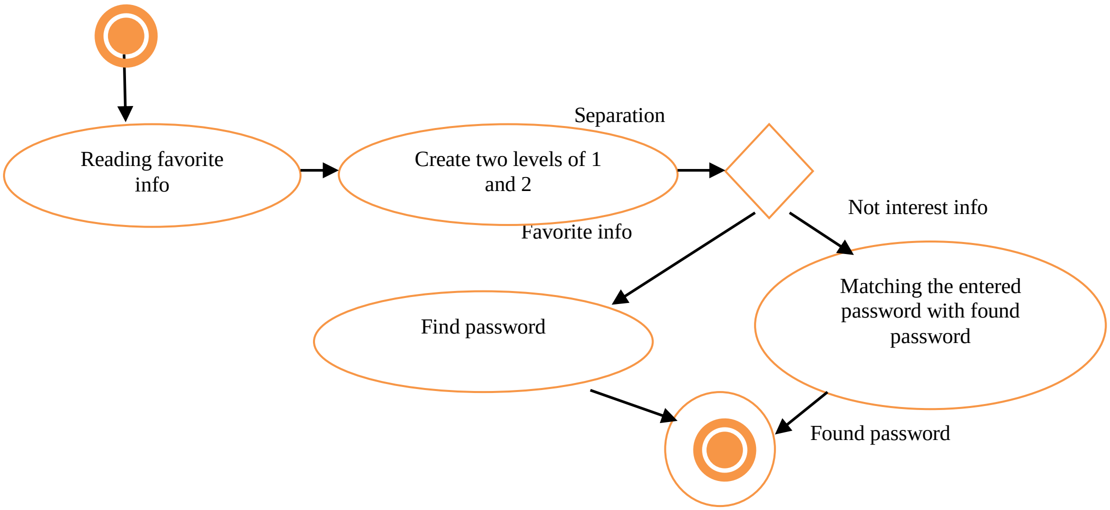

Figure 16. Planner model of classification agent.

Since capability of updating agent is modeled as storage and recovery agent by interaction model, there is no need to use capability and programming models for this agent.

- Service model

The service model is determined using the rules and constraints section of the knowledge model in the analysis phase. In this model every role is associated with at least one service. For every service, inputs, outputs, preconditions and post conditions should be determined. Inputs and outputs are easily extracted from the interaction model. Table 13 shows the service model for the real estate trade system.

Table13. Service model of house buying & selling system.

Service name	Input	Output	Preconditions	Post conditions
Listing real estate information	Seller or buyer ID no	The estate information display	Inserting the usage license	The history time the register.
Information update	New information insertion	Information reforms	Information insert for edition	Event time register for history and saving editions in data bank.
Information ordering	Favorite / un favor information	Illustration of information saved in list 1/2	Information type determination	Time registration for events in the history of the list 1/2 in the information bank.

The proposed methodology is based on the results from the real estate trade system in experts view point with respect to these three methodologies and performed a good support for parameters like autonomy, objective orientation interaction ability, and domain usability. In expert's point of view, models added to the combinational proposed model leads to an increase in convergence between the analysis and design phases. According to the view of the experts due to this modals ARA methodology is suitable for analysis and designing of business and industrial systems. Determination of agent capabilities and presentation of capabilities with these modals is important because of these systems. Analyzing team from the beginning of the project was able to determine the details accurately and the failure of this project has been minimized. Taking in consideration that the procedure of capabilities modal presentation and program is a new procedure, these two models can be used in other engineering software agent oriented methodologies and increase the quality of these methodologies.

6. Conclusion and future work

In this study a developed combinational methodology was used for analysis and design of agent- oriented systems. In this methodology, by combining strengths of ASPECS, ROADMAP and AOR methodologies and adding actor, capability and programmer models, it is possible to use high-level techniques to manage the problem complexity. Using the combinatory method in the proposed methodology led the achievement of two main goals of working standards and redefinition of the main blocks. The proposed methodology in the expert viewpoints and in the form of evaluation parameters in agent-oriented engineering has a good support for parameters like autonomy, goal orientation, react ability and domain usability compared to other methodologies.

Although three methodologies were selected for evaluation and formalization, they aren't a complete to show of all agents' methodologies. There are many important AOSE methodologies that each of them has special different features to support different aspects of operational dominion goals. Therefore future works can focus on expanding selected methodologies and using current evaluation framework to evaluate them. By doing this, uniform stated by models and techniques can be increased. Also by considering potential risks related to quality evaluation used in this article, other studies can be done on quantity experiences

References

[1] Zambonelli, F., Jennings, N. R., & Wooldridge, M. (2005). Multi-agent systems as computational organizations: the Gaia methodology. Journal of Autonomous Agents and Multi-Agent Systems, vol. 9, no. 3, pp. 136-171.

[2] Weiß, G. (2001). Agent orientation in software engineering. The knowledge engineering review, vol. 16, no.4, pp.349-373.

[3] Tveit, A. (2001). A survey of Agent-Oriented Software Engineering, In Norwegian University of Science and Technology, (pp.104-125), 2001.

[4] Garcia-Ojeda, J. C. & DeLoach Robby, S. A. (2009). AgentTool Process Editor: Supporting the Design of Tailored Agent-based Processes, In Procedings of the ACM Symposium on Applied Compting, (pp. 707-714). ACM, 2009.

[5] DiLeo, J., Jacobs, T., & DeLoach, S. A. (2002). Integrating Ontologies into Multiagent Systems Engineering. 4th international bi-conference workshop on agent- oriented Information systems, (pp.23-34), 2002.

[6] DeLoach, S. A. (2014). O-MaSE: An Extensible Methodology for Multi-agent Systems. In Agent-Oriented Software Engineering, vol. 4, no. 3, pp. 173-191.

[7] DeLoach, S. A., & Garcia-Ojeda, J. C. (2014). The O-MaSE Methodology. In Handbook on Agent-Oriented Design Processes, (pp. 253-285), 2014.

[8] Juan, T., Pearce, A., & Sterling, L. (2002). ROADMAP: Extending the Gaia Methodology for Complex Open Systems. The First International Joint Conference on Autonomous Agents & Multi-Agent Systems, pp. 3-10, 2002.

[9] Zambonelli, F., Jennings, N. R., Omicini, A., & Wooldridge, M. J. (2001). Agent-oriented software engineering for internet applications. In Coordination of Internet Agents, pp. 326-346, 2001.

[10] García-Ojeda, J. C., Arenas, A. E., & Alcázar, J. P. (2005). Paving the Way for Implementing Multiagent Systems: Integrating Gaia with Agent-UML. 6th International Workshop Agent-Oriented Software Engineering. Lecture Notes in Computer Science 3950, pp. 179-189, Springer-Verlag, 2005.

[11] Gonzalez-Palacios, J., & Luck, M. (2007). Extending Gaia with Agent Design and Iterative Development. 8th International Workshop Agent-Oriented Software Engineering, LNCS 4951, pp 16-30, 2007. Springer-Verlag, 2007.

[12] DeLoach, S. A. (2002) Modeling Organizational Rules in the Multiagent Systems Engineering Methodology, Proceedings of the 15th Canadian Conference on Artificial Intelligence, pp.54-62.

[13] Giorgini, P., Mylopoulous, J., & Sebastiani, R. (2005). Goal- Oriented Requirements Analysis and Reasoning in the Tropos Methodology. J. of Engineering Applications of Artificial Intelligence, Elsevier, vol. 8, no.2, pp. 159-171.

[14] Jureta, J., Faulkner, S., & Schobbens, P. (2006). Allocating Goals to Agent Roles During MAS Requirements Engineering. 7th Int. Workshop Agent-Oriented Software Engineering, LNCS 4405, pp. 19-34, Springer-Verlag, 2007.

[15] Cossentino, M., Gaud, N., Hilaire, V., Galland, S., & Koukam, A. (2009). ASPECS: an agent-oriented software process for engineering complex systems. J. of Autonomous Agents and Multiagent Systems Systems, vol. 20, no. 2, pp. 260–304.

[16] Wagner, G. (2003). The Agent-Object-Relationship Metamodel: Towards a Unified View of State and Behavior. J. of Information Systems. vol. 28, no. 5, pp. 475-504.

[17] Odell, J. (2002). Objects and agents compared. Joural of object technology, vol. 1, no. 1, pp. 41-53.

[18] Lind, J. (2001). Issues in agent-oriented software engineering. In Agent Oriented Software Engineering (pp. 45-58). Springer Berlin Heidelberg, 2001.

[19] Dam, K. H., & Winikoff, M. (2004). Comparing agent-oriented methodologies. In Agent-Oriented Information Systems, pp. 78-93, Springer Berlin Heidelberg, 2004.

[20] Sturm, A., & Shehory, O. (2004). A framework for evaluating agent-oriented methodologies. In Agent-Oriented Information Systems, pp. 94-109, Springer Berlin Heidelberg,2004.

[21] Ghandehari, E., Saadatjoo, F., & Chahooki, M. A. Z. (2014). AMA: a compound methodology for designing and implementing agent-based systems. J. of Advances in Computer Science, vol. 3, no. 11, pp. 107-114.

[22] Sudeikat, J., Braubach, L., Pokahr, A., & Lamersdorf, W. (2005). Evaluation of agent–oriented software methodologies–examination of the gap between modeling and platform. In Agent-Oriented Software Engineering, pp. 126-141, Springer Berlin Heidelberg,2005.

[23] Hsu, Y. L., Lee, C. H., & Kreng, V. B. (2010). The application of Fuzzy Delphi Method and Fuzzy AHP in lubricant regenerative technology selection. Expert Systems with Applications, vol. 37, no. 1, pp. 419-425.

[24] Bauer, B., Müller, J. P., & Odell, J. (2001). Agent UML: A formalism for specifying multiagent software systems. International Journal of Software Engineering and Knowledge Engineering, vol. 11, no. 3, pp. 207-230.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

ادغام متد: روشی به منظور توسعه و بهبود متدولوژی‌های عامل‌گرا

عرفان قندهاری[۱]، فاطمه سعادت‌جو[۱*] و محمدعلی زارع چاهوکی[۲]

[۱] دانشکده مهندسی کامپیوتر، دانشگاه علم و هنر، یزد، یزد، ایران.

[۲] دانشکده برق و مهندسی کامپیوتر، دانشگاه یزد، یزد، یزد، ایران.

ارسال ۲۰۱۴/۰۷/۴؛ پذیرش۲۰۱۵/۰۵/۰۷

چکیده:

مهندسی نرم‌افزار عامل‌گرا یکی از زمینه‌های جدید و در حال توسعه‌ی علوم کامپیوتر است که در قالب متدولوژی‌های عامل‌گرا، رویکردهای سیستماتیک برای تحلیل، طراحی، پیاده‌سازی و نگهداری سیستم‌های چندعاملی ارائه می‌دهد. در بحث مهندسی نرم‌افزار عامل‌گرا یکی از چالش‌های مهم و اساسی این است که علی‌رغم متدولوژی‌های متعددی که در این حیطه معرفی شده، هنوز کاستی‌هایی در فازهای مختلف متدولوژی‌ها وجود دارد. در این مقاله با توجه به نقاط ضعف متدولوژی‌های عامل‌گرا در فازهای مختلف و تأثیر به‌سزایی که این کاستی‌ها، می‌تواند در کیفیت و کارایی پروژه‌های نرم‌افزاری داشته باشد به ارائه راهکاری ترکیبی برای معرفی متدولوژی (ARA) از متدولوژی‌های ASPECS، ROADMAP و AOR پرداخته‌ایم. بدین منظور سه متدولوژی مورد بررسی در قالب یک چارچوب ارزیابی جامع بر اساس معیارهای مفاهیم و ادراک‌ها، زبان مدل‌سازی، فرآیند و عمل‌گرایی مورد ارزیابی قرار گرفته‌اند. با توجه به محدودیت‌های منابع و زمان، تصمیم به انتخاب متدولوژی‌های نمونه برای ارزیابی و ادغام اتخاذ گردید. این انتخاب بر اساس میزان کاربرد متدولوژی‌های مورد بررسی و همچنین امکان ادغام آن‌ها از دید افراد متخصص و خبره صورت گرفته است. بر اساس نتایج ارزیابی، متدولوژی ROADMAP مراحل تحلیل سیستم‌های عامل‌گرا را به‌خوبی پشتیبانی می‌کند و مرحله‌ی طراحی در آن به دلیل عدم مدل کردن تعامل بین عامل‌ها، کامل نمی‌باشد، از طرف دیگر با توجه به این که متدولوژی‌های ASPECS و AOR مراحل طراحی و تعامل‌های بین عامل‌ها را به‌خوبی پشتیبانی می‌کنند، با استفاده از مزایای سه متدولوژی، یک متدولوژی ترکیبی ارائه می‌گردد که در آن از مرحله‌ی تحلیل متدولوژی ROADMAP و مرحله طراحی متدولوژی‌هایASPECS و AOR استفاده می‌شود. همچنین به منظور افزایش کارایی متدولوژی پیشنهادی مدل‌های کنشگر، سرویس، قابلیت و برنامه‌ریز نیز به مدل‌های موجود در متدولوژی پیشنهادی اضافه شده است و به منظور تشریح فازهای مختلف آن، در یک مطالعه‌ی موردی، مورد استفاده قرار گرفته است. نتایج این پژوهش می‌تواند زمینه را برای معرفی نسل بعدی متدولوژی‌های عامل‌گرا فراهم آورد.

کلمات کلیدی: مهندسی نرم‌افزار عامل‌گرا، سیستم مبتنی بر عامل، ASPECS، ROADMAP، AOR.