Towards Intelligence Engineering in Agent-Based Systems

Shiva Vafadar and Ahmad Barfourosh
Computer Engineering and IT Faculty, Amirkabir University of Technology, Iran

Abstract: *In this paper, we consider intelligence as an explicit requirement in modern information systems and introduce intelligence engineering as the application of a systematic and disciplined approach to the development, operation, and maintenance the intelligence. After categorizing intelligence sub-characteristics based on an extensive quantitative analysis, we choose learning as the most frequent sub-characteristic of intelligence and present a process for specifying learning in intelligent systems. This process is based on Organizational Multiagent System Engineering (O-MaSE) process framework and extends this framework to cover Learning Analysis (LA) concepts. More precisely, by using O-MaSE meta-model and LA meta-models, alongside applying the process for development a system, we study the potentials of the process framework in specifying learning concepts. Based on the discovered shortcomings of the process, we propose extensions to enrich the process for specifying learning, namely for Requirements gathering, Goal modeling, Knowledge modeling and Environment specification. The applicability of the extended O-MaSE is evaluated by applying it on a Book Trading System (BTS). Based on this systematic process, software development benefits from the more specific assumptions on capabilities expected for learning, and it moves towards more sophisticated engineering practices which prevents implicit or Ad-hoc activities for developing features of intelligent.*

Keywords: *Intelligence, learning, O-MaSE, intelligence engineering, agent oriented software engineering.*

Received May16, 2012; accepted June 28, 2012, published online April 17, 2014

1. Introduction

Due to the complex nature of modern software systems, intelligence is a feature of advanced applications for handling this complexity e.g., [12]. In order to make intelligence part of the everyday practices of software development, software engineers need techniques and tools to specify intelligence. Specification of intelligence features, similar to the other features, is a key factor for increasing the quality of the modern software systems.

We introduce intelligence engineering based on IEEE definition on software engineering [10] as the application of a systematic and disciplined approach to the analysis, design, implementation, test, operation and maintenance the intelligence. This paper moves forward our research for specifying intelligence as an explicit feature of software systems. In the previous works, to clarify intelligence, we presented a framework for precise definition of intelligence requirements of software systems. Presented framework is based on a well-known idea about intelligent that states "Intelligence is not a single mental process, but rather a combination of many mental processes…" Encyclopedia Britannica, 2006. Consequently, this framework represents the taxonomy of intelligence, which categorizes the sub-characteristics of intelligence via a quantitative analysis on more than 70 definition of intelligence [14].

Based on this study, learning is the most frequent sub-characteristic of intelligence. Learning is a capability incorporated to software systems in order to help them perform more appropriately in dynamic and volatile situations. In the next step of our research, we presented a meta-model of learning which is used for analysis of the learning, as a feature of software systems [14].

The next step, which makes the idea more feasible and it is the main concern of the present work, investigates the potentials of the existing software development processes for specifying learning. Using Learning Analysis (LA) meta-model, the aim of our research is twofold. First, we study the strengths and weaknesses of a selected process for developing the learning requirements. Second, we investigate how the general model of the LA meta-model is joined and applied in the process. This approach evaluates the applicability of the meta-model as a supplementary document for developing the learning in software development process.

In this paper, the scope of the work is Agent Oriented Software Engineering (AOSE). AOSE is a relatively new paradigm in software society that makes Artificial Intelligence (AI) findings much closer to the other fields of science and engineering [17]. Based on our investigation and evaluation on the existing agent-oriented processes and methodologies [8], we choose the Organizational Multiagent System Engineering (O-MaSE) process framework [3, 4, 5, 7]. O-MaSE is an appropriate candidate because it covers requirements

gathering, problem analysis and solution analysis activities. These activities are the main steps of requirements engineering [9] which are not properly addressed in current research, despite of design and implementation techniques of learning [6, 16]. In addition, O-MaSE presents a meta-model which contains the main concepts used to define multi-agent systems in this framework. This meta-model is properly fit to the LA meta-model.

Based on meta-model and experimental evaluation techniques [1] first, we compare and adapt the learning and O-MaSE meta-models and then apply O-MaSE for developing a Book Trading System (BTS). According to the evaluation, we report the discovered strengths and shortcomings of the process framework for specifying learning. Next, we extend the framework to overcome the discovered shortcomings. Consequently, we introduce a development process based on the extended O-MaSE for specifying learning in intelligent systems. The applicability of the extensions is evaluated by developing the BTS via using extended O-MaSE and proposed process.

The remainder of this paper is organized as following: Section 2 explains the meta-models of learning. Section 3, introduces the O-MaSE process framework and it's meta-model. In section 4, the meta-models are mapped and the discovered strengths and shortcomings of O-MaSE are discussed. In section 5, extensions are introduced to enrich O-MaSE for specifying learning. Section 6 introduces our process for developing learning and its application in BTS. Finally, section 7 contains the conclusion and further works of our research.

2. Learning Analysis

In this section, we introduce the leaning analysis meta-models briefly. More detailed information is presented in [14, 15]. This set of meta-models contains a group of related, generic meta-classes of learning and their relations in a domain neutral manner, which can be described as elements of conceptual modeling of learning requirements of agents. Figure 1 shows an abstract meta-model of learning.

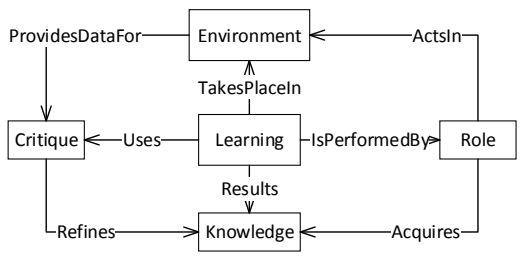

Figure 1. Learning meta-model.

This model indicates that, in an abstract view, learning is performed by a learner role in an environment that provides data for critique. This critique is used to refine the knowledge, which is acquired by the learner. To have more detailed specification of the learning, the concepts of role, critique, Environment and knowledge are described in the following, respectively.

Role is a concept of the learning, which represents the learner. Figure 2 shows the learning role meta-model. This model indicates that role is a learner such as human or agent that has specific goals. To analyze the learning, a specific goal should be defined which is called learning goal. Learning goal helps the agent to achieve its goal. The learning role performs specific Tasks to meet its learning goals. These tasks vary from exploration of new circumstances to exploitation of the previous findings.

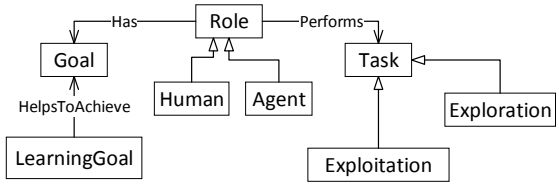

Figure 2. Role meta-model.

Critique is another concept of the learning, which represents the criticism available for evaluating the learner's performance. Figure 3 represents the core concepts of the critique during learning. It indicates that the learner uses feedback from the critic to determine how its actions should be modified to act better in the future. According to this model, performance measure is defined as a fixed standard for evaluating how well the role performs. For measuring the learning, a specific type of measure is defined which is called Learning Measure. It is also necessary to identify the available feedback and the way it is obtained for the learner. In the learning process, critique observes performance measure, evaluates learning measure, and creates feedback, as the model shows.

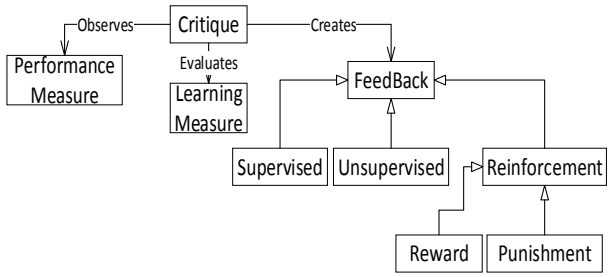

Figure 3. Critique meta-model.

Environment is another important concept of the learning. How well, an agent behaves and learns partially depends on the nature of the environment. The core concepts of the environment for the learning are shown in Figure 4. This model indicates that, properties of the environment, laws that restrict perceptions and interactions, in addition to the data which is provided by the environment are the important issues for describing the environment of the learning.

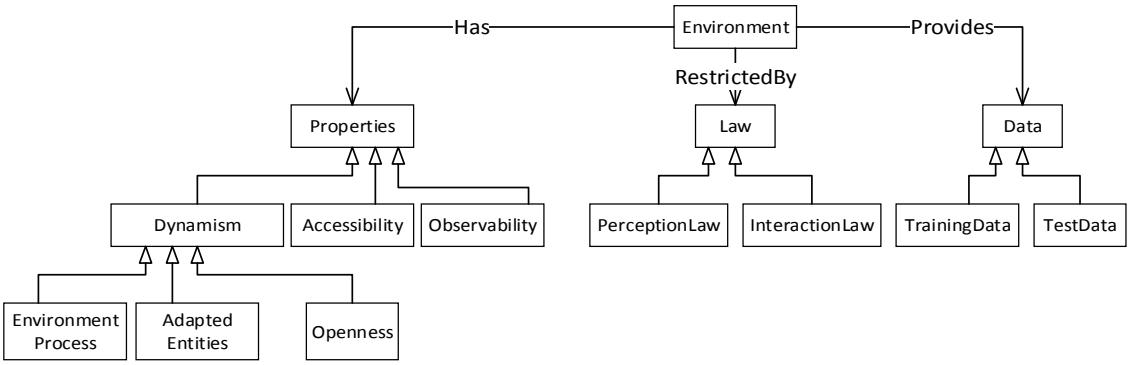

Figure 4. Environment meta-mode.

Acquiring the knowledge is usually the ultimate goal of any learning process. Figure 5 represents the main concepts of the knowledge which is used, generated or refined during learning. The knowledge specification is started by identifying the learning subject. Two types of knowledge should be elicited for learning: Pre-Knowledge, which is available for the learner before it starts its work and acquired knowledge, which are knowledge elements that the agent captures during the learning.

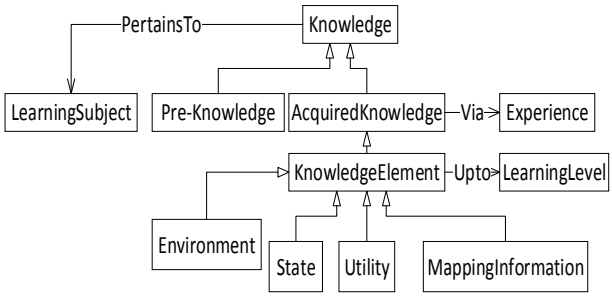

Figure 5. Knowledge meta-model.

3. O-MaSE

O-MaSE [3, 4, 5] is a process framework for constructing custom agent-oriented processes. It introduces a set of methods fragments that can be combined to obtain customized processes. O-MaSE identifies six main activities: Requirements gathering, problem analysis, solution analysis, architecture design, low level design and code generation. For each activity, it introduces a set of tasks, techniques and work products.

O-MaSE is based on a common meta-model, which defines a set of concepts and their relations that are used in the developing of agent-based systems. Figure 6 shows this meta-model. According to the model, the organization of agents contains five concepts; goals, roles, agents, domain model (including environment object and environment property), and policies. This meta-model also consider Actors, protocols and capabilities composed of actions and plans, as important concepts in multi-agent systems.

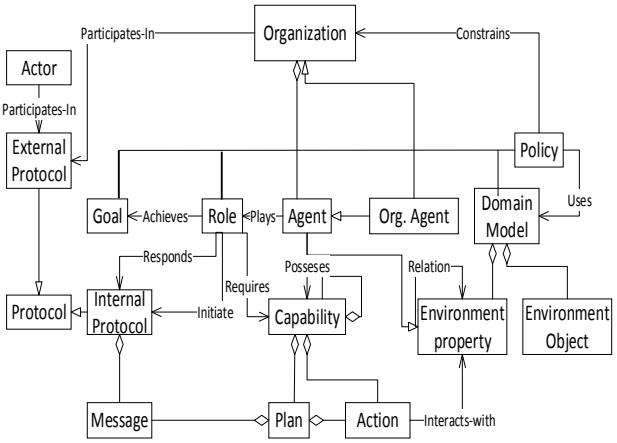

Figure 6. O-MaSE meta-model.

4. Evaluating O-MaSE for Specifying the Learning

In this section, we compare and map meta-models of LA Figures 1, 2, 3, 4, and 5 with the O-MaSE meta-model Figure 6. The purpose is to distinguish which issues of the learning specification are covered by the O-MaSE framework. This indicates the potential of O-MaSE and its models for specifying the learning.

Tables 1, 2, 3, and 4 show the results of this mapping. These tables also show how these concepts are operationalized in a learning process by extending the models or instantiating them for learning specifically. In the following, we discuss the relationship between these meta-models in detail.

Table 1. Role and O-MaSE meta-models mapping.

LA	Role	Goal	Learning Goal	Task
O-MaSE	Role	Goal	Softgoal	Action
O-MaSE/Learning Process	Learner	Goal	Learning Softgoal	Explore Action, Exploit Action

Table 2. Critique and O-MaSE meta-models mapping.

LA	Critique	Performance Measure	Learning Measure	Feedback	Supervised	Unsupervised	Reinforcement	Reward	Punishment
O-MaSE	-	Acheive	Acheive	-	-	-	-	-	-
O-MaSE/ Learning Process	Critique Protocol/ Plan	Performance Acheivement	Learning Acheivement	Feedback Protocol/ Plan	Supervised Protocol/ Plan	Unsupervised Plan	Reinforcement Protocol/ Plan	Reward Message	Punishment Message

Table 3. Environment and O-MaSE meta-models mapping.

LA	Environment	Properties	Dynamism			Accessability	Observability	Laws		Data
			Environmental Process	Adaptive Entity	Openness			Interactional	Perceptional	
O-MaSE	Domain Model	Environment Properties	-	-	-	-	-	Policy		Message, Hard capabilities
O-MaSE /Learning Process	Domain Model	Environment Properties	Environment Description	Environment Description	Environment Description	Environment Description	Environment Description	Policy		Learning Data

Table 4. Knowledge and O-MaSE meta-models mapping.

LA	Knowledge	Learning Subject	Pre-Knowledge	Acquired-Knowledge	Knowledge Element	State	Mapping Information	Utility	Environment	Experience	Learning Level
O-MaSE	-	Environment Object Type	-	-	-	-	-	-	-	Action	-
O-MaSE/ Learning Process	Knowledge Model	Learning Subject	Knowledge Model	Knowledge Model	Knowledge Model	Knowledge Model	Knowledge Model	Knowledge Model	Knowledge Model	Explore Action	Learning Level

The closest concept to learning in LA is capability in O-MaSE. Both LA and O-MaSE meta-models represent the concept of role. In LA, the role is defined more specifically and indicates that this role can be a human or an agent that its learning capability is specified. Goal and learning goal in LA represent the concept of goal and soft goal in O-MaSE, respectively. In both models, identifying the goals is the first task of the analysis. Learning is a capability that helps to achieve the goal(s). In LA, learning goal emphasizes on the relationship between learning and goals. It denotes the improvements in the goal achievement expected to be obtained by incorporating the learning. Task in LA and action in O-MaSE, similarly, specify the behavioral elements to perceive or act upon the environment. LA denotes that these tasks are exploration and exploitation for generating and using experience.

Although, there is not a critique concept in O-MaSE, it can be defined via protocols and plans as shown in Table 2. In OMACS [4], achieve is a function that defines how well a role achieves a goal. This evaluation can be performed via defining performance measure and learning measure during learning specification. It is defined to evaluate how well a role achieves the learning goal.

According to Table 3, O-MaSE specifies the environment by domain model (containing environment object type and environment property). In LA, it is considered as the environment meta-model which specifies the properties of the environment that should be captured for LA such as dynamism, accessability, observability. Data is defined as any input that is received from the environment and used as the input of learning process. In O-MaSE, messages in protocols and hard capabilities such as sensors provide the data.

As Table 4 shows, learning subject is captured in O-MaSE, as an environment object type, and action is mapped to experience for generating knowledge. In this table, knowledge and its related elements, such as knowledge elements and learning level are not mapped to a concept of O-MaSE explicitly.

4.1. Discussion

In order to evaluate O-MaSE for learning specification, in addition to, meta-model evaluation, we use experimental evaluation via applying it in a BTS due to the space limit, these artifacts are not presented in this paper. Our study on learning specification in O-MaSE shows some strengths and weaknesses in this process framework. We discuss them here and provide some extensions to overcome the weaknesses in the next section.

1. O-MaSE meta-model covers some of the required concepts of the learning, based on the concepts defined in LA models as shown in Tables 1, 2, 3, and 4 this indicates that various models of the O-MaSE framework that represent these concepts i.e., goal model, domain model, organizational model, role model, agent class model, protocol model, policy model, Plan model and capability model, are applicable for modeling the learning feature. Our experiment in modeling the BTS confirms this mapping, as well. However, we recognize that extensions that are needed for focusing on the learning-related issues, especially in the earlier activities and tasks of the process are beneficial, which are provided in the next section.
2. In O-MaSE, requirements of the system are captured via using traditional or new multi-agent approaches [3]. Currently, the expectations from agent intelligence are not defined explicitly in agent society. This hinders engineers to consider intelligence from the first steps of software development, which increases the risk of implicit or Ad-hoc activities for dealing with intelligence in later phases. According to our approach, learning is considers as a requirement of the system. Consequently, it should be mentioned in requirements specification of the system. In the next section, we present a boilerplate that supports this perspective.
3. Despite of the other agent oriented methodologies such as ASPECT [2] O-MaSE does not consider knowledge as a concept in agent-based systems. Although, it is mentioned that the domain model is appropriate for modeling the knowledge of the domain [3] the agent knowledge is not necessarily related to the domain or the environment objects. This knowledge can be about mapping from the

conditions of the current state to the actions, desirability of the world state or desirability of the actions [11]. Since, learning is a process that acquires or refines the knowledge of the agent, the available pre-knowledge and acquired knowledge during the learning should be identified in the solution analysis activity.

4. Learning goal is a soft or quality goal of the system and it needs to be defined in Goal model. Although, soft goals are supported in a T3 tool [7], developing soft goals is considered as further extensions in O-MaSE [11]. For modeling intelligence-related requirements, this feature is required. The relationships between goals also need to be extended. Since, the relationship between learning and other goals is not necessarily "Precedes" or "Trigger", in this task a new type of relationship is needed.
5. In O-MaSE, environment property is part of the domain model. But, O-MaSE documents and examples do not address how these properties are captured. Since, environment and its properties plays a vital role in learning algorithm selection, it should be specified explicitly in learning specification. In the next section, we present a template for environment documentation.
6. O-MaSE introduces a framework for generating customized processes. By using the learning meta-model as a guiding framework, it is distinguished that the software development process for learning specification should contain goal modeling, domain modeling, organizational modeling, role modeling, agent modeling, protocol modeling, policy modeling, capability modeling, and plan modeling from original O-MaSE. This process also needs extended models, namely requirements specification, soft goal modeling, environment specification and knowledge modeling which are described in the following.

5. O-MaSE Extensions

In this section, we present the required extensions to overcome the discovered weaknesses for modeling the learning in O-MaSE. In the next section, we show how these extended models and the original O-MaSE models are combined to generate a software development process for learning.

5.1. Learning Requirements Specification

As we consider learning as a requirement of the system, it should be specified in Software Requirements Specification (SRS) document. In O-MaSE, there is not a stipulated technique. Therefore, we propose using the boilerplate presented in Table 5 for specifying learning requirements. In this template, the phrases in < > are defined in each application that its learning requirement is specified.

Table 5. Learning requirement specification template.

Summary	<Learning Aspect> **Learning**
Definition	The system (or specific **<Role>)** shall learn **<Aspect>** since <**Initiator**> within **<Life Time>** (Within **<Duration>** **<Unit>** or after receiving <**Amount of Data**> < **Input Data**>) in **<Environment>** When **<Precondition>** The <**Output**> is shown to the user and it is refreshed <**Automatic Refreshing>** The learning should produce the results with **<Precision>** The motivation of this requirement is <**Motivation>**

5.2. Goal Modeling Extension

In goal modeling, learning is modeled as a soft or quality goal. Usually there is one or more goals in the goal model that learning helps to acheive them better in terms of the time or quality. In this task, Attribute-Precedes-Trigger (ATP) [6] technique is used to show the relationship between the learning soft goal and other goals that are affected by it. Since, the relationship between learning and other goals is not necessarily "Precedes" or "Trigger", we extend the relationship by adding "Contribute" relation to it. This relation is used to illustrate how learning contributes to the other goals of the system, both positively or negatively.

5.3. Environment Specification

In Domain modeling, the properties of the environment should be specified. Important environment properties for specifying the learning are observability, accessability, and dynamism. These properties are defined during analysis and are used in detailed design to select the most appropriate learning algorithm, based on the environment conditions. Table 6 provides a template for learning environment specification.

Table 6. Environment description document template.

0-		**System Environment**	**Cause**
Accessability			
Observability			
Dynamism	Environmental Process		
	Adaptive Entities		
	Openness		

5.4. Knowledge Modeling

In knowledge modeling task, the goal model and domain model of the system is used as the input. The domain model is constructed by modeling the knowledge related concepts, predicates and actions. To accomplish this task, the domain object types that are related to learning soft goal (e.g., learning subject) are selected from the domain model. This core model is enriched by appending the knowledge elements that are available or required to achieve the learning soft goal. These knowledge elements are predicates that can be asserted about the status of these concepts and the actions that affect the status of the concept.

Finally, in this model the pre-knowledge elements are distinguished from acquiring concepts during learning for example by different colors. This model is used to generate knowledge base of the system during the design activity.

6. Customized Process for Learning Specification

In this section, we show that how the extended O-MaSE is applied for modeling a BTS which exploit learning as one of its features. In this regard, we develop the case study with emphasize on its learning capability, by following the proposed process.

BTS is an open multi-agent system supporting the market of various books requiring the interaction between individuals. In this market, the sellers introduce their products and intended prices. Buyers call for the books while they know the maximum price that can be afforded. There are brokers that introduce the sellers and buyers to each other. Sellers can change the book price based on the market condition. Therefore, the sellers should learn the best price that increases the profit.

6.1. Requirements Specification

In this activity, the requirements of the system are specified in a SRS document. In BTS, learning is a requirement of the seller that needs an intelligent learner agent for selling books. Based on our proposed template in Table 5, this requirement is defined as shown in Table 7.

Table 7. Book's price learning requirement in BTS

Summary	Book's Price Learning
Definition	The SellerAgent shall learn the BestPrice of each book since the selling books is started, within no longer than 1 week (or 10000 bargain) in a dynamic environment. In this environemt there are other seller and buyer agents added or deleted to the system. The learnt price is shown to the user and is refreshed daily. This learning should be performed with 85% preciseness. The motivation is to increase the Seller profit 10%.

6.2. Problem Analysis

The first task of the LA, is defining the goals of the system. These goals are identified by using the requirements of the system. In goal modeling, learning is modeled as a softgoal based on the learning requirements. The relationship between the learning soft goal and other goals are specified via preceed, trigger and contribute relations. Figure 7 shows part of the extended and refined goal model of BTS.

Domain model is used to specify the object types that agent interact and reason about. In domain modeling, the entities or the concepts of the domain and the environment are depicted. Object types are used to specify goal and event parameters, message parameters, constraints and the result of agent actions.

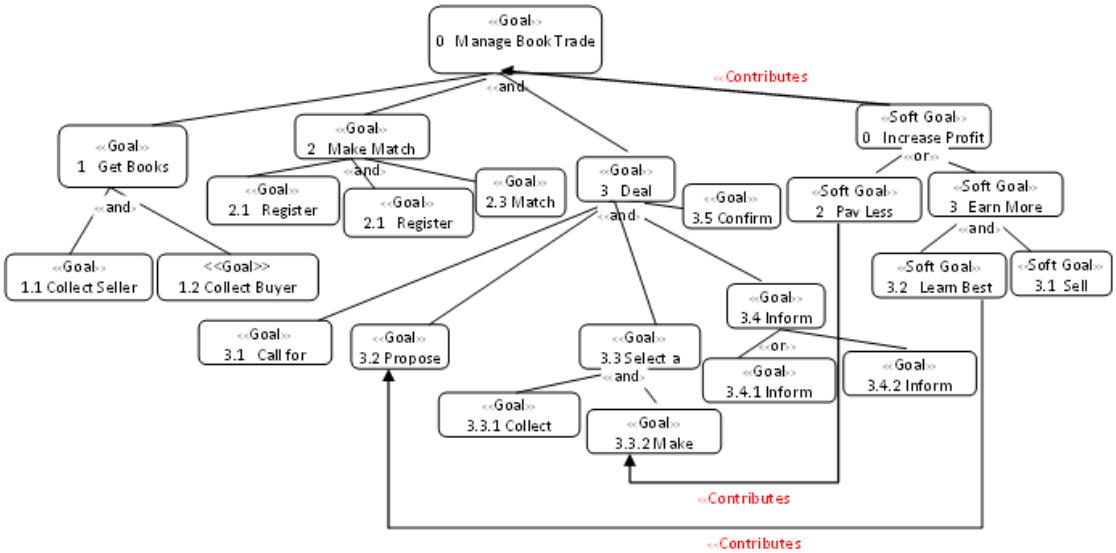

Figure 7. Extended goal model of dynamic system for BTS.

The domain model of the BTS as shown in Figure 8, which environment object, agents and the messages are transmitted in the system.

During Domain modeling, the specific properties of the environment for learning are described. In BTS, the environment is accessible for the seller agent, because it can bid various prices and collect information about them. It is partially observable, since the results of the bids are announced but the seller agent does not have any model of other agents, and it is open because other seller and buyer agents are added or deleted from the agent organization as described in Table 8.

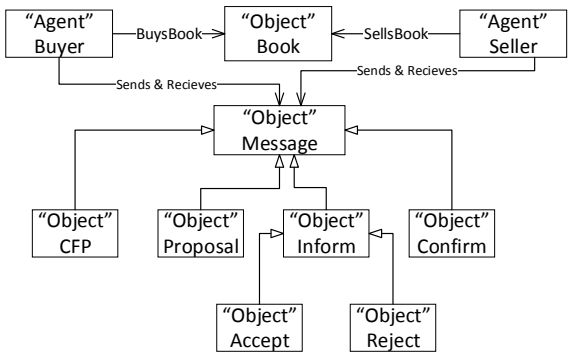

Figure 8. Domain model of BTS.

Table 8. Environment description document for BTS.

Environmet Property	BTS Envionment	Cause
Accessability	Accessible	Exploring various Prices
Observability	Partially Observable	Observable: Deal winner Non-observable: Other agents' model
Dynamism	Environmental Process	Business Trend
	Adaptive Entities	Competing Sellers
	Openness	Sellers Agents, Buyer Agents

6.3. Solution Analysis

The aim of the Organization model is to identify the system's interfaces with external actors. For supervised or reinforcement learning, the relationship between external actors and organization should be modeled in this task. In the BTS, the system interacts with the buyers and sellers as the actors. In this system, the human user provides the feedback for the seller agent as complementary input data for the learning. In Figure 9, train protocol specifies this relationship.

Role modeling is another task for solution analysis. The focus of this model is on identifying the roles required in the organization and their interactions via defining protocols. Each role should achieve at least one leaf goal of the goal model. In a system with learning feature, a "Learner" role is defined, which is responsible for achieving the learning soft goal, as shown in Figure 10 for BTS.

Figure 9. Organizational model of BTS.

Knowledge modeling is another task for solution analysis. This task captures the pre-knowledge and acquired knowledge for achieving the learning soft goal. This model enriches the knowledge related objects of domain model by adding concepts, predicates (relation between concepts) and the actions that affect the concepts and relations. In BTS, learning subject is book, which is an environment object.

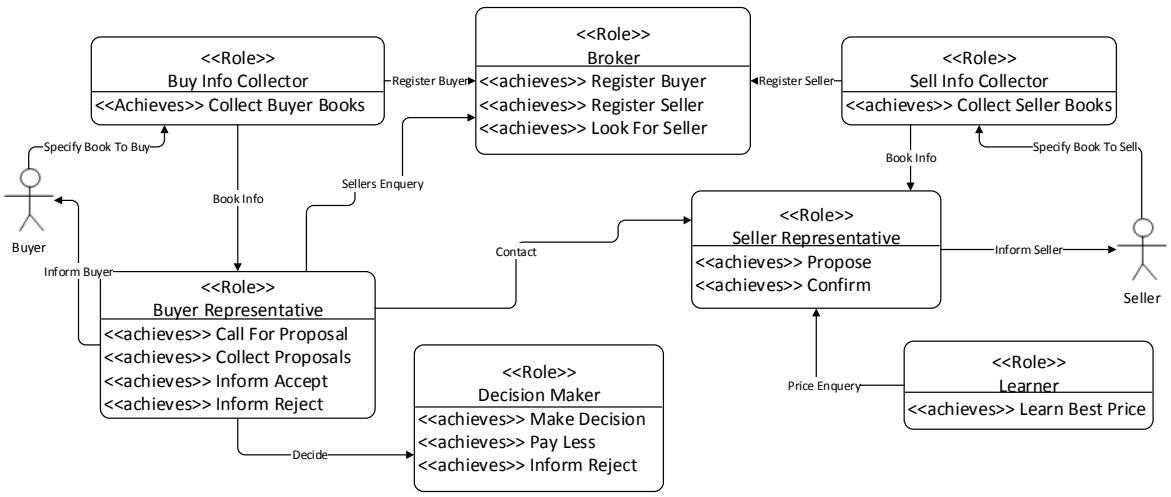

Figure 10. Role model of BTS.

The seller agent receives the book price as its initial knowledge and refines it by learning the concept of desirability of various prices for the book in the market condition. A portion of the Knowledge model of learning agent is shown in Figure 11. In this model, the pre-knowledge concept is "Price" and knowledge concepts that should be acquired are "Trend" and "Price desirability" and required action is "Trial".

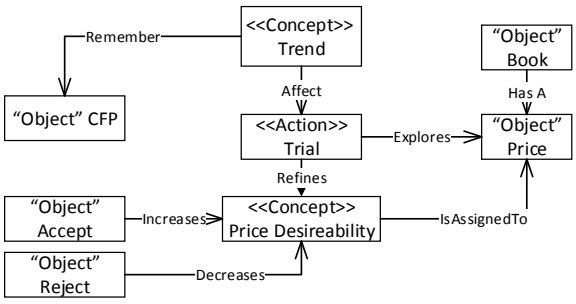

Figure 11. Learning knowledge modeling of BTS.

In defining roles, learning is defined as a capability and is assigned to the "learner" role. A portion of role description document of BTS is shown in Table 9. In this task, plans are defined to make the capabilities operational.

Learning capability is realized via a plan, which is modeled in low level design. Notice that while other roles access to the database, learner role accesses to the knowledge base.

Table 9. Role description document.

Role	Acheives	Capability	Plan
Buyer info collector	Collect Buyer Book	User Interaction, Database Access	-
Buyer representative	Call For Proposal, Collect Proposal, Inform Seller	Communication Database Access	-
Decision maker	Make Decision	Database Access	
Seller representative	Propose, Confirm	Communication, Database Access	-
Learner	Learn Best Price	Learning, Knowledgebase Access	Learn Price

6.4. Architecture Design

In this activity, the first task is Agent classes modeling. In this task, agent classes are defined to play specific roles or they are specified in terms of capabilities. At least an agent plays "Learner" role. This agent is entitled to be called "Intelligent", because it possesses a sub-characteristic of intelligence [13]. In BTS, the seller agent, which plays "Learner" role and possesses "Learn" capability is the intelligent agent of the system. Figure 12 shows the Agent class model of BTS.

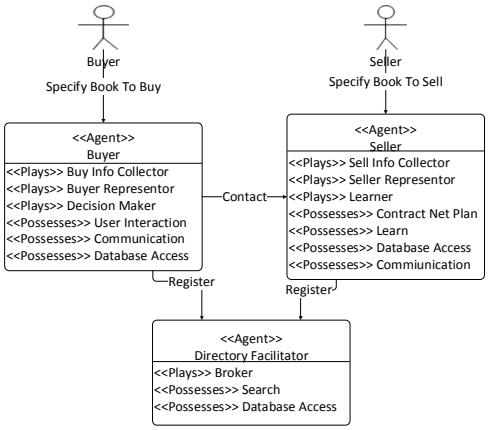

Figure 12. Agent model of BTS.

Architecture design activity also includes Protocol modeling. The purpose is to define the details of the interactions between agents or roles. When learning is performed via reinforcement or supervise learning, the interactions between the "Learner" role and critic (the actor or internal agent) are modeled in this task.

In BTS, the learning related protocols are "Contract" and "Train" Protocols. Figure 13 shows the Contract protocol, which is a reinforcement learning protocol. This protocol provides the required input data (containing rejection or acceptance of seller proposals) for acquiring the knowledge about desirability of the price by seller.

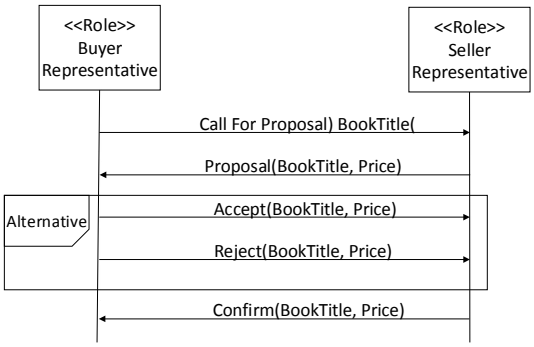

Figure 13. Contract protocol (reinforcement learning protocol).

Figure 14 shows the Train protocol, which is a supervised learning protocol in BTS. In this protocol, user (via seller info collector role) provides learning data for the seller.

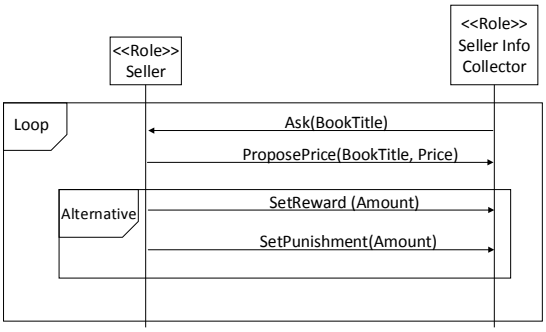

Figure 14. Train protocol (supervised learning protocol).

6.5. Low Level Design

Capability modeling task is used to define the capabilities possessed by the agents. Each capability is modeled as an action or plan. Learning is defined as a capability which is performed by a plan. Due to the space limit capability model of BTS is not presented here.

In O-MaSE plan is an algorithmic definition of the capability that uses actions and implements protocols. In BTS, this plan model is used to define the learning soft capability as shown in Figure 15. In an intelligent system, access the knowledge base is another capability based on Table 9, which is modeled in low level design activity.

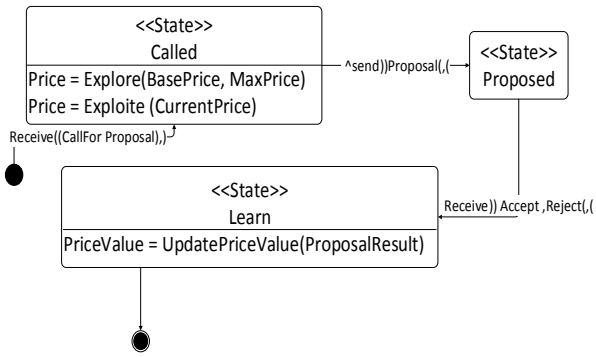

Figure 15. Learn price plan.

7. Conclusions and Further Work

In this paper, we moved one step forward in our research towards intelligence engineering. In this way, we presented a development process for learning, as the most frequent sub-characteristic of intelligence. The process is based on O-MaSE process framework, which is extended for learning specification. The extensions were suggested based on the comparison of learning meta-models with O-MaSE meta-model and a practical experience in developing the learning via O-MaSE. The extended process contains some original O-MaSE models, i.e., goal model, domain model, role model, agent model, protocol model, policy model, capability model, and plan model in addition to the extended new models, namely requirements specification, refined goal model, knowledge model and environment specification. Based on this

systematic development process, learning is specified from the first activities of software development and it contains more specific assumptions on expected capabilities for the learning. This prevents implicit or Ad-hoc activities for developing learning and leads software development to a more sophisticated engineering practices.

The further work of our research includes investigating the potentials of agent oriented methodologies and processes for specifying other sub-characteristics of intelligence, such as planning and reasoning. Our final goal is providing a meta-model of intelligence and enriching the existing agent-oriented methodologies to cover it. This will enable AOSE to support intelligence engineering for systematic development of intelligence, which is a demand in modern information systems.

References

[1] Brinkkemper S., Hong S., and Van G., "A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies," *in Proceedings of the 26th Hawaii International Conference on System Sciences*, vol. 4 , pp. 689-698, 1993.

[2] Cossentino M., Guad N., Hilaire V., Galland S., and Koukam A., "An Agent-Oriented Software Process for Engineering Complex Systems," *Journal of Autonomous Agents and Multiagent Systems Systems*, vol. 20, no. 2, pp. 260-304, 2009

[3] DeLoach S. and García-Ojeda J., "A Customisable Approach to Designing and Building Complex, Adaptive Multi-Agent Systems," *International Journal Agent-Oriented Software Engineering*, vol. 4, no. 3, pp. 244-280, 2010.

[4] DeLoach S. and Miller M., "A Goal Model for Adaptive Complex Systems," *International Journal of Computational Intelligence: Theory and Practice*, vol. 5, no. 2, pp. 1-20, 2010

[5] DeLoach S., Oyenan W., and Matson E. "A Capability Based Model for Artificial Organizations," *Journal of Autonomous Agents and Multiagent Systems*, vol. 16, no. 1, pp. 13-56, 2008.

[6] Garcia A., Kulesza U., Sardinha J., Milidiu R., and Lucena C., "The Learning Aspect Pattern," *avalible at: http:// hillside.net/plop/2004/ papers/ agarcia0/PLoP2004_agarcia0_0.pdf*, last visited 2004.

[7] Garcia-Ojeda J., DeLoach S., and Robby, "Agent Tool Process Editor: Supporting the Design of Tailored Agent-based Processes," *in Proceedings of the ACM Symp on Applied Compting*, New York, USA, pp. 707-714, 2009.

[8] Henderson-Sellers B. and Giorgini G., *Agent Oriented Methodologies*, idea Group Publishing, 2005.

[9] Hull E., Jackson K., and Dick J., *Requirements Engineering*, Second Edition, Springer, USA, 2005.

[10] IEEE Standards Collection: Software Engineering., available at: https://standards.ieee.org/findstds/prod/cds/softeng.pdf, last visited 1993.

[11] Russell S. and Norvig P., *Artificial Intelligence: A Modern Approach*. 2nd Edition, Prentice Hall, 2003.

[12] Turanin A. and Al-Shrouf F., "Intelligent e-Learning Framework for Practicing Educational Testing Services," *the International Arab Journal of Information Technology*, vol. 8, no. 4, pp. 372-375, 2011.

[13] Vafadar S. and Abdollahzadeh A. "Characteristics of Computational Intelligence Quantitative Approach," *in Proceedings of the 22nd Midwest conference Artificial Intelligence and Cognitive Science*, Cincinati, USA, 2011.

[14] Vafadar S. and Abdollahzadeh A., "Towards Requirement Analysis Pattern for Learning Agents," *in Proceedings of the 11th International Conference on Agent-oriented Software Engineering*, pp. 79-90, 2011.

[15] Vafadar S. and Abdollahzadeh A., "Analysis Patterns for Learning Agents," *in Proceedings of the 15th European Conference on Pattern Languages of Programing*, Germany, 2010.

[16] Van H., Hennes D., Kaisers, M., Tuyls K., and Verbeeck K., "Multiagent Learning Dynamics: A Survey," *in Proceedings of the 11th International Workshop*, Delft, The Netherlands, vol. 4676, pp. 36-56. 2007.

[17] Zambonelli, F., Omicini, A." Challenges and Research Directions in Agent-Oriented Software Engineering," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 9, no.3, pp. 253-283, 2004.

Shiva Vafadar is currently an assistant professor in Golestan University, Gorgan, Iran. She was a PhD student in Computer Engineering and IT Department of Amirkabir University of Technology, Iran. She was a visiting student in FBK research center, Trento, Italy (2011). Her research interests are: software engineering, intelligence engineering, agent oriented software engineering, agent oriented methodologies, intelligence requirement patterns, and intelligence analysis patterns.

Ahmad Barforoush is a professor in Computer Engineering and IT Department of Amirkabir University of Technology, Iran. He was a visiting professor in Maryland College Park (USA), ORASY (Paris 11), Trento (Italy) and Loughborough (UK) during 20 years ago. He is the author of books entitled “Introduction to Distributed Artificial Intelligence” and “Software Quality Assurance Methodology”. His research areas are: artificial intelligence, agent-based systems, automated negotiation, expert systems, natural language processing, decision support systems, business intelligence, data mining, data warehouse and software engineering.