Journal of Engineering Science and Technology
Vol. 18, No. 5 (2023) 2655 - 2671
© School of Engineering, Taylor's University

ONTOLOGY-BASED KNOWLEDGE ENGINEERING FROM VARIOUS AGENT SYSTEMS

ARDA YUNIANTA[1,*], SHAMINI RAJA KUMARAN[2], ANAS ABULFARAJ[1], AHMAD HOIRUL BASORI[1], ANDI BESSE FIRDAUSIAH MANSUR[1], OMAR MOHAMMED BARUKAB[1]

[1]Faculty of Computing and Information Technology in Rabigh,
King Abdulaziz University, Jeddah, Saudi Arabia
[2]Department of Artificial Intelligence, School of Electrical
Engineering and Artificial Intelligence, Xiamen University Malaysia
*Corresponding Author: ayunianta@kau.edu.sa

Abstract

Many organizations use multi-agent systems as a system solution to deal with complex problems in their organization. However, there are still obstacles to solving semantic aspect problems related to the data and information in different agents' data sources in the multi-agent system. Semantic aspect problems are about data with different names but having similar meanings or data with similar names but having different meanings. In the previous works, the researchers proposed new methodologies, named "Ontology-based Methodology for Multi-Agent Systems (OmMAS)". OmMAS is to develop ontology knowledge to solve semantic aspect problems for multi-agent systems. However, there still needs proof of implementing OmMAS to solve semantic aspect problems in the real-life case study. This research aims to test and implement OmMAS with a real case study to solve semantic aspect problems in the learning agent's domain and demonstrate how we can extract knowledge from different agents' data sources. Furthermore, this research presents the knowledge as a result of ontology development and also analyses and evaluates the advantages and weaknesses of the OmMAS. In this research, there are two main parts presented. The first part is ontology development process to show how to extract the knowledge from learning agents' data sources using OmMAS. It produces ontology knowledge with a semantic relationship scheme to solve semantic aspect problems. The second part is the result of ontology development and discussion to analyse and evaluate the OmMAS. In addition, from the result and discussion section, we mentioned some advantages and weaknesses of OmMAS as an improvement for future work.

Keywords: Education domain, Knowledge engineering, Methodology, Multi-Agent system, Ontology development, Semantic approach.

1. Introduction

The development of systems and applications has become more complex and faces many aspects of problems, especially for multi-agent system development [1-3]. A few years ago, a multi-agent system was developed as a solution to deal with problems such as information exchange between systems, handling large amounts of data, and complexity in systems and application development [1, 4]. Multi-agent systems have several features that are very useful for managing and controlling complex applications and systems [5, 6]. The first feature, as the main focus of a multi-agent system, is autonomous, which means a multi-agent system takes decisions without being explicitly told by a human being.

The second feature is intelligent interaction and collaboration in the systems and applications environment [1, 2], and the third feature is that multi-agent systems are able to self-adapt at runtime to any unexpected events. The fourth feature is mobility, which means the ability to move from one system to another system with easy adaptability and many other features of multi-agent systems [7-11]. However, in recent years, the problems in multi-agent systems have become increasingly complex, especially in the methodological steps to overcome semantic issues [12-14].

Several semantic issues in the multi-agent system arise in the relationships between data or information in the multi-agent system. The relationships between data in different agent systems produce data conflicts because there is no semantic mapping between data that has other names with the same meaning and data that have the same name but a different meaning [15, 16]. For example, in the education agent system, there is a possibility to store data about undergraduate and postgraduate students in different databases and agent systems with the same table names as a student.

Another possibility is that data about the same doctors are stored in different databases and agent systems with different table names, in one agent system with a table name as an instructor and in another agent system using the name as a lecturer. In recent years, ontology implementation for multi-agent systems has been used to try to solve these semantic issues in the multi-agent system development process.

The methodology of multi-agent system development is meant to produce a decent implementation of the agent system [17-19]. There are a lot of existing methodologies for multi-agent system development, and every methodology has different objectives and specific goals to overcome certain problems in a particular domain [20-24]. However, this research focuses on OmMAS (Ontology-Based Methodology for Multi-Agent Systems) as the latest methodology in the development of a multi-agent system using the ontology approach.

OmMAS is also our contribution to the previous research as one of the solutions to developing agent systems to solve semantic aspect problems in the current multi-agent system implementation [13]. The second contribution is an improvement in several processes from previous ontology-based methodologies on multi-agent systems [12]. Nevertheless, the previous paper's research about OmMAS still didn’t perform the testing and implementation for the OmMAS with a real case study to recognize the strengths and weaknesses in the OmMAS development phases.

The aim of this research is to test and implement OmMAS with a real case study in the education domain, and subsequently is to identify and analyse the strengths and weaknesses of OmMAS. To achieve this goal, we divided this paper into two main parts. The first part of this paper is to detail the Ontology-Based Methodology for Multi-Agent Systems (OmMAS) and implement and experiment with every process of OmMAS using real case studies on the Learning Agent domain. The second part of this paper is the main contribution of this research; this section presents the Ontology development result as a solution for semantic aspect problems in the multi-agent system.

Furthermore, from the implementation of OmMAS, we analyse and evaluate OmMAS to provide some advantages and weaknesses of OmMAS for future improvement. All explanations in the result and discussion section will be a continuous improvement of OmMAS as a background problem to perform the following research in the future and to get a better methodology phase to develop a multi-agent system in the future. The paper is organized as follows: Section 1 covers the introduction, continued by Section 2, the research methodology that explains the whole process of the OmMAS model. Section 3 focuses on the results and discussion, and finally, section 4 is the conclusion.

2. Research Methodology

Ontology-based Methodology for Multi-Agent Systems (OmMAS) [13] is an improvement of the previous [12, 25] ontology development methodology for a multi-agent system. As shown in Fig. 1, there are a total of ten phases in the OmMAS, with three additional phases as a contribution from previous work, and the other seven phases are the common ontology development phases that have been used in the existing ontology development methodologies.

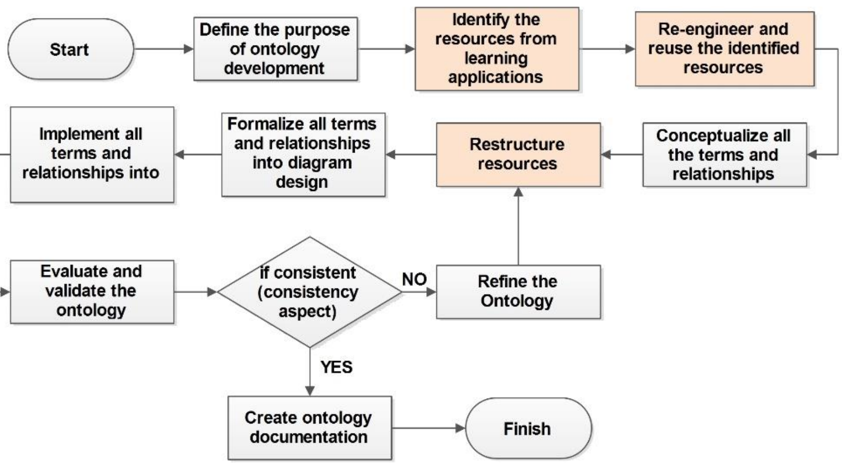

Fig. 1. Ontology development phases in the OmMAS.

The general phase to develop the ontology is important to define the purpose of ontology development, and this process is the first phase of OmMAS. The second phase is resource identification from all agent systems. The third phase is reengineering activity and reusing all important information in agent systems. The fourth phase is to conceptualize all the terms and relationships that will be used in the ontology. The fifth phase of OmMAS is to restructure and reorganize all terms and relationships to be formal ontology knowledge [26].

The sixth phase is to design and draw all terms and relationships. This phase is the conceptual model to transform all terms and relationships into a semi-computable model. The seventh phase is to develop ontology knowledge and implement all terms and relationships using a specific ontology development tool. After the ontology development process, the eighth phase is the evaluation and validation of ontology knowledge with a consistent acceptance level. The evaluation and validation activity involves an iterative process to refine the ontology to get consistency-level acceptance in the ontology knowledge [27]. This process is the ninth phase of OmMAS. The last phase, the tenth phase in the OmMAS, is to finish the documentation of ontology development.

2.1. Define the purpose of ontology development

Every development process has an objective, and it is important to formulate this at the beginning of ontology development. The objective helps a developer to set the coverage of the ontology developed [28]. The aim of ontology development in this research is to produce learning ontology knowledge that is able to collaborate with various learning agents in the education domain.

2.2. Identify the resources from multi-agent system

Identification of the resources is the next step of the development process. The purpose of this phase is to find any possible data and information related to the learning information between different resources [29]. There are two different data learning agents and two types of learning content or information that want to be linked between these two data sources as shown in Fig. 2.

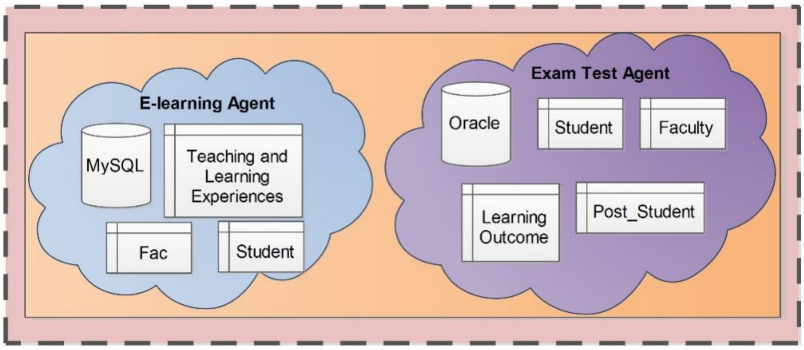

Fig. 2. Resources of multi-agent system.

These two agents, called e-learning and exam test agents, contain a lot of learning content. However, this research only focuses on seven learning contents that are related and have a semantic aspect problem. Furthermore, all these seven learning contents are representation of database tables in every agent system. In the e-learning agent system, there are three learning contents namely teaching and learning experiences, faculty, and student, while in the exam test agent system there are four learning contents namely learning outcome, faculty, student, and post student.

2.3. Reengineer and reuse the multi-agent resources

The reengineering and reusing of resources phase are a crucial phase because this process is to selects important information in the learning agents. In these two data sources shown in Fig. 2, there are a lot of data schemas that represent tables on data sources. There are a lot of existing tables on every data source, and there are many possibilities to produce data conflict, data duplication, semantic problems, and other data problems [30]. However, from all the tables on the data sources, only specific tables are selected that contain information about teaching and learning experiences and learning outcomes.

The representation of information from two data sources has different schemas designed. Two semantic aspects occur among these two resources. The **first semantic aspect** that happens between these two resources is about different name in these two tables that has the same information contained in that table. The Question Bank agent uses the Fac name with two fields, but the E-learning agent uses the Faculty name with three fields to store the same information about faculty. So, in summary, the Fac table name in the E-learning system agent and the Faculty table name in the Question Bank system agent have the same meaning as the different table names and data representations because these two tables contain the same information (meaning).

The **second semantic aspect** is about the different data with the same representation name in these two agents. The E-learning system agent uses student name representation to store data about all students (undergraduate students and postgraduate students), while the Question Bank system agent use the same name namely student to store the data about undergraduate students only. So, in summary, the student table name in the E-Learning system agent and the student table name in the Question Bank system agent have different meanings for the same table name because these two tables contain different information (meaning).

2.4. Conceptualize all the terms and relationships

The aim of this step is to generate and reform all possible terms and relationships to be meaningful models from the ontology perspective. All possibilities of information selected from agent systems are to be considered and become terms. In other words, all information in every table in the databases will extract to become terms, because we will reorganize every information that already become terms and find the relationship between them. For example, in the data source, there are several data/tables related to students' data. Still, from these two agent systems (presented in Fig. 2), there are different representations to save the data related to students' data.

Therefore, we create the terms Student, StudentUndergaduate and StudentPostgraduate to accommodate the information/data from every data source from these two agent systems. In this phase, the domain ontology conceptual model should be developed [31]. All terms in this step are transformed into classes and subclasses from the ontology perspective. Every class and subclass represent information from the ontology perspective, and any information contained in every class and subclass is called an individual or instance. The possibilities of all terms are shown in Table 1.

After all terms described, the next step is to define semantic relationships that occur between all terms in this ontology knowledge. Basically, all semantic relationships are from table relationships between tables in the database. Still, we need to identify and describe more possibilities for every (semantic) relationship between all terms because in this ontology we combine two different data sources in two different agents. Semantic relationships bridge all of the information in resources to create more valuable information called knowledge. Semantic relationships bridge all information inside each resource and all information outside resources (between each resource in different agent system). The possibilities of all relationships are shown in Table 2, while the possibilities of all data properties are shown in Table 3.

Table 1. List of terms.

No.	Terms	No.	Terms
1	CourseLearningOutcomes	15	LearningPerson
2	StudentUndergraduate	16	Wiki
3	ProgrameLearningOutcome	17	Forum
4	SubjectCoursePostgraduate	18	Survey
5	SubjectCourseUndergraduate	19	Workshop
6	WeeklyScheduleActivities	20	Resources
7	WeeklySchedulePlans	21	Lecturer
8	StudentPostgraduate	22	Student
9	LearningOutcomes	23	Department
10	SubjectCourse	24	CourseOutline
11	WeeklySchedule	25	Faculty
12	Weeks	26	Semester
13	Presentation	27	Major
14	LearningActivities		

Table 2. List of relationships.

No.	Relationships	No.	Relationships
1	hasProgramLearningOutcome	7	hasLecturer
2	hasAssessmentMethod	8	hasAnActivities
3	hasNumberOfPresentation	9	teaches
4	hasCourseLearningOutcome	10	enrols
5	hasLearningTimeName	11	performs
6	hasLearningTime	12	conductedIn

Table 3. Data properties.

No.	Data Properties	No.	Data Properties
1	weeklySchedulePlanCode	13	lecturerId
2	subjectCourseCode	14	forumCode
3	weeklyScheduleAcitivitiesCode	15	semesterCode
4	presentationCode	16	facultyCode
5	courseLearningOutcomesCode	17	forumCode
6	programLearningOutcomeCode	18	surveyCode
7	departmentCode	19	icNumber
8	workshopCode	20	idNumber
9	resourcesCode	21	weeksCode
10	postalAddress	22	studentGrades
11	emailAddress	23	phoneNumber
12	wikiCode		

2.5. Restructure resources

The primary purpose of the restructure resources phase is to try to detect missing knowledge, make some improvements, and finally reorganise the knowledge contained in the initial conceptual model [25]. All data properties will connect to the data/information in the database, and finally we can extract all data from the tables in the databases into data properties.

In this phase, ontology developers should restructure the resources to modify all terms to be classes and subclasses, then identify the suitable data properties for a specific class or subclass. Table 3 describes the possibilities of data properties used in ontology. Not every instance has data property, but only a few essential instances have data property. While Table 4 shows the details of the restructuration of terms into classes and subclasses in the ontology.

Table 4. Restructure all terms into classes and sub classes in ontology perspective.

No.	C	SC	Terms
1	✓		LearningActivities
2		✓	Resources
3		✓	Wiki
4		✓	Forum
5		✓	Survey
6		✓	Workshop
7		✓	Presentation
8	✓		Semester
9	✓		Major
10		✓	Faculty
11		✓	Department
12	✓		SubjectCourse
13		✓	SubjectCoursePostgraduate
14		✓	SubjectCourseUndergraduate
15	✓		LearningPerson
16		✓	Student
17		✓	Lecturer
18		✓	StudentPostgraduate
19		✓	StudentUndergraduate
20	✓		CourseOutline
21		✓	WeeklySchedule
22		✓	WeeklySchedulePlans
23		✓	Weeks
24		✓	WeeklyScheduleActivities
25		✓	LearningOutcomes
26		✓	CourseLearningOutcome
27		✓	ProgrameLearningOutcome

*C: Classes; **SC: Sub Classes

2.6. Formalize all terms and relationships into diagram design

Diagram design is a formalization process of all relationships and terms into an ontological concept. The purpose of this phase is to design and draw all the ontological concepts defined in the previous step to be a semi-computable model

[13]. This phase involves some activities and specific tools to help make it into a diagram design. This design will help with the ontology development activity in the next step. The design for the ontology diagram is shown in Fig. 3.

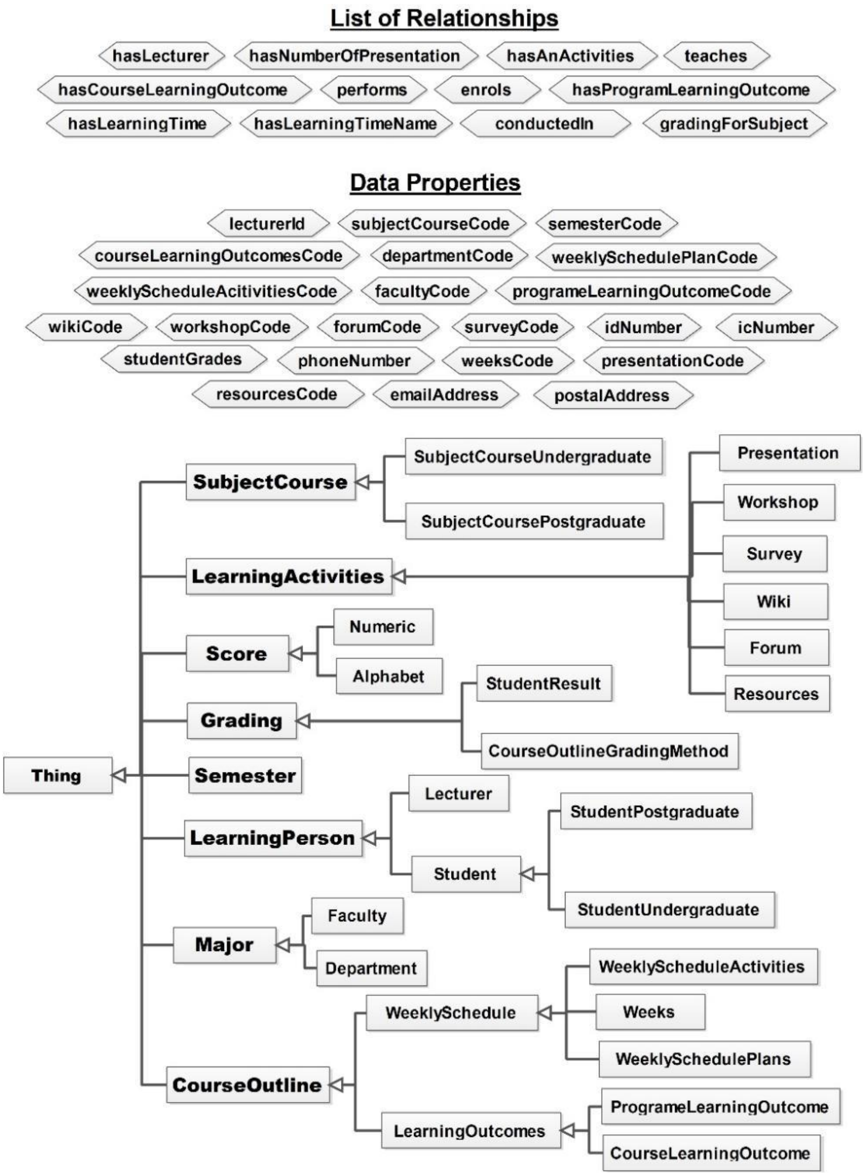

Fig. 3. Design ontology diagram.

2.7. Implement all terms and relationships into ontology

Implementing all terms and relationships into the ontology phase is an ontology model knowledge construction. A development of ontology knowledge is to design an appropriate ontology representation model that can be saved into a standard ontology language that can be used for other programming languages [13]. Figure 4 shows the ontology knowledge result exported from the Protégé 4.3 tools.

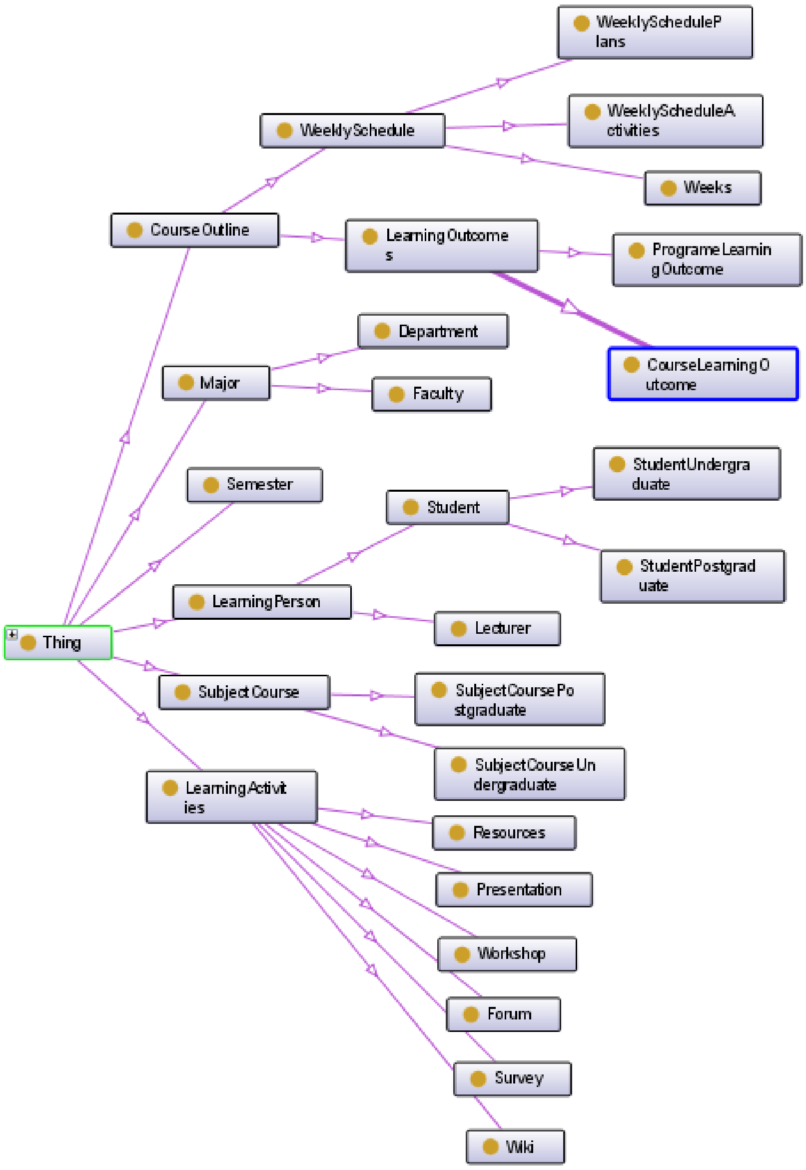

Fig. 4. Ontology knowledge.

This phase is the core process, and it is very important to represent the ontology model more precisely and to be unambiguous. In this research, the implementation of all terms and relationships is done using specific tools. Protégé 4.3 is chosen as a tool to implement all terms and relationships needed to be an ontology knowledge base. The Protégé tool is one of the recommended and famous tools to develop ontology knowledge. The main reason for this research is to use Protégé, which is a free tool and has a reasoner as one of the methods to evaluate and validate the ontology knowledge.

2.8. Evaluate and validate the ontology

There are semantic aspects that need to be considered and solved in ontology knowledge. Ontology consistency is very important and needs to be evaluated and validated. Every time an inconsistency is found in the ontology knowledge, that time also needs an improvement process. The evaluation and validation process will fulfil the consistency level in the ontology knowledge consistency acceptance result [32]. The ontology has two options to validate it.

The first option is using the W3 standard directly through www.w3.org as a standard validation method to validate the ontology RDF file format [27, 32]. The second option is using reasoners in the 3.4 tools. Reasoner is one of the features included in the protégé tools [33, 34]. Ontology validation using protégé 3.4 tools and the W3 standard is to measure the consistency of learning ontologies. Figure 5 shows the example view if the ontology still has some inconsistency errors.

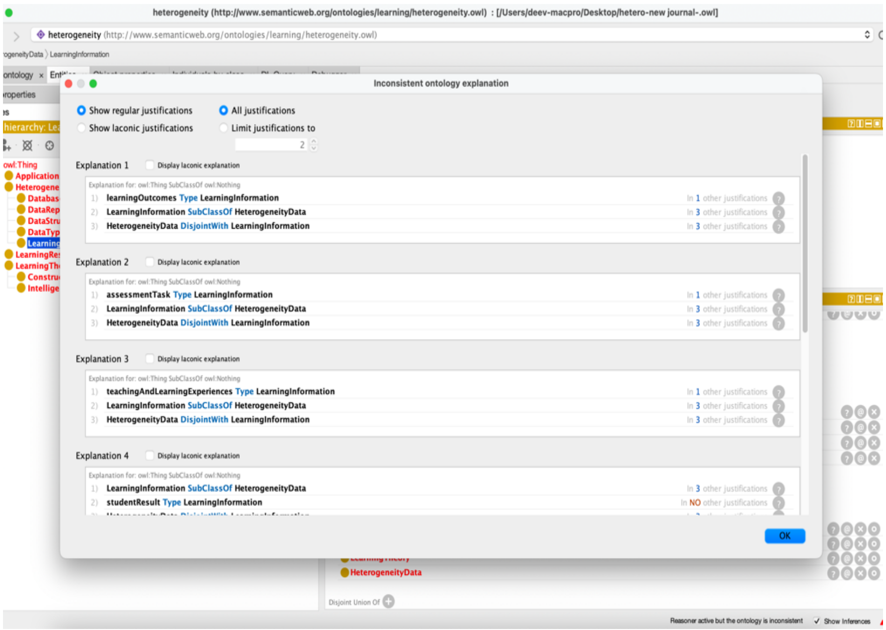

Fig. 5. Inconsistency ontology result.

2.9. Refine the ontology

Refining the ontology phase is to fix, edit, and improve the ontology knowledge that has an error report in the semantic consistency aspect [35]. The refinement process stops when there is no more inconsistency error report, but as long as it still has an inconsistency problem in the ontology knowledge, the iterative process will continue until the ontology knowledge gets acceptance results in the evaluation and validation process.

2.10. Create the ontology documentation

The ontology documentation file is very important for ontology clients or future developers. All processes in the ontology development must be completed and documented to help the client or future developers to maintain and reuse the ontology or make any improvements in the future.

3. Result and Discussion

The result of ontology development is the knowledge obtained by adding relationships between the information from each data source contained in the data sources of different agent systems. The data properties (shown in Table 3) in the ontology take all the data from the data source, and the list of relationships (as shown in Table 2) connects all the related data into knowledge. The difference between the relationships in databases and ontologies is the number of data relationships that can be built and combined to obtain meaningful data. A database can only establish a relationship between two different types of data, whereas an ontology can connect two or more types of data to form knowledge. Table 5 shows some examples of knowledge that can be obtained from the results of the ontology development performed in the previous section.

All knowledge is created by combining multiple data sources from data sources that are connected by ontological relationships. All the data retrieved from the data source is stored in the data properties of the ontology, and then all these data properties are combined into knowledge using ontology relationships. Table 5 shows that there are multiple data properties and ontology relationships in a knowledge statement. Thus, the purpose of knowledge creation is to represent all the data contained in the data source in such a way that it can be transformed into knowledge that is easier to understand from a human perspective.

Table 5. Examples of knowledge from ontology development.

No.	List of Knowledge's	List of Relationships	List of Data Properties
1	Richard enrols IT123 conducted in sem12021.	Enrols ConductedIn	Richard = idNumber IT123 = subject-CourseCode sem12021 = semesterCode
2	IT123 teaches by Mr Ahmad has course learning outcome A, B and C.	Teaches HasCourse-LearningOutcome	IT123 = subject-CourseCode Mr Ahmad = lecturerId A, B and C = courseLearning-OutcomesCode
3	Course IT123 has learning time Monday and Tuesday.	hasLearningTime	IT123 = subject-CourseCode Monday and Tuesday = weekly-SchedulePlanCode
4	Course IT123 has an activities Course materials, forum activities, and students' presentation.	hasAnActivities	IT123 = subject-CourseCode Course materials = resourcesCode Forum activities = forumCode Students' presentation = presentationCode

This ontology knowledge result shows how ontology can solve the semantic aspect problem mentioned in Section 2.3. In ontology knowledge, we reconstruct and reorganize all information from all data sources and reform them into meaningful information. The solution for the first semantic aspect is to avoid the duplication of data from different data sources with other names with the same information contained in data sources. While the solution for the second semantic aspect problem is that we can recognize the difference in the information between two data sources with the same representation name. We solve both semantic aspect problems by conceptualizing all terms and relationships mentioned in section 2.4 and refining them with activities in subsequent processes.

Table 6 presents the comparison of our proposed approach with similar previous work. There are four main factors: Methodology, Information/Data, Semantic problems, and solutions for semantic issues. In terms of methodology, our proposed method focused on the multi-Agent system in the E-learning system, while others focused on the financial and supply chain management systems. The data that we used cover all attributes of the e-learning system. Regarding semantic problem handling, the proposed method can handle data with different names but containing the same information, then different data with the same representation.

Table 6. Comparison with other studies.

Studies	Methodology	Information/ Data	Semantic problems	Solution for semantic problems
An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage (2015) [4]	Kacktus	Learner/user, Course	Firstly, to incorporate e-learning systems effectively in the evolving semantic web environment and secondly, to realize adaptive personalization according to the learner's changing behaviour	Creating ontology knowledge that interact with agent system
Designing an ontology-based multi-agent system for supply chain performance measurement using graph traversal (2014) [25]	MOBMAS: A Methodology for Ontology-Based Multi-Agent Systems Development	Suppliers, manufacturers, and customers	to achieve the interoperability in the supply chain (SC) complex and heterogeneous environment.	new method based on integrating the agent technology as well as the ontology technology by designing new architecture
A Methodology for Creating Ontology-Based Multi-Agent Systems with an Experiment in Financial Application Development (2013) [12]	the MOMA methodology	Score, Working Capital/Total Assets, Retained Earnings/Total Assets, Earnings Before Interest & Tax/Total Assets, Market Value of Equity/Total Liabilities, Sales/Total Assets	1. They are able to describe macroscopic properties of a system already in existence, but not the origin of these properties. 2. They cannot be easily applied to situations where the assumptions behind mathematical equations no	Creating ontology knowledge that interact with agent system

			longer hold. 3. They do not handle heterogeneity in populations well.	
This study	Ontology-based Methodology for Multi-Agent Systems (OmMAS)	Course Learning Outcomes, Learning Person, Student Undergraduate, Wiki, Programe Learning Outcome, Forum, Subject Course Postgraduate, Survey, Subject Course Undergraduate, Workshop, Weekly Schedule Activities, Resources, Weekly Schedule Plans, Lecturer, Student Postgraduate, Student, Learning Outcomes, Department, Subject Course, Course Outline, Weekly Schedule, Faculty, Weeks, Semester, Presentation, Major, Learning Activities	**first semantic aspect** that happens between these two resources is about different name in these two tables that has the same information contained in that table. **second semantic aspect** is about the different data with the same representation name in these two agents.	Creating ontology knowledge that interact with agent system

One of the goals on this work is to test and implement OmMAS in various domains and case studies in order to gain a better understanding of its performance. The methodology phases of OmMAS consist of complex and many phases, but on the other hand, it makes the development process a long journey and it takes a long time to complete the development process. The complexity of the development phase is great for creating documentation files, and this will be very helpful for the development of multi-agent systems in the future because the developers will get a better understanding and better view of the ancient multi-agent system.

In the first phase of the OmMAS, how to define the development objectives of the ontology, it is important to know the purpose of the development process and it is also important to identify the limitations of the knowledge and information coverage on ontology. Determining the purpose of the development process also means that it tries to identify resources and chooses (restricts) important information involved in the development process, and both of these processes are also related to the second and third phases in OmMas. Therefore, we conclude that these two phases can be combined into one phase.

The documentation phase is to create file documentation to record all phases in the development process. We argue that every process in the development steps is the documentation itself, because from the beginning we create a document to finish every step in the development process. We agree that the documentation step in the development process is very important, but if we put the documentation step in the final step, it is not entirely true because we create documentation files since we start from the first development process.

In conclusion, we note that there are several phases in OmMAS that can be simplified to make the development process shorter and more efficient. Moreover, we can combine several phases and redesign the OmMAS phase to provide a better view and better methodological steps to improve OmMAS and make it more effective and efficient in developing ontologies in multiagent systems.

4. Conclusion

Semantic aspect problems in various domains of multi-agent systems are still engaging and need to be addressed. One of the solutions for semantic aspect problems is to use an ontology development approach that has been proven in many studies. In this research, we have succeeded in overcoming the problem of semantic aspects using OmMAS in the learning domain agent system. There are four significant processes in the OmMAS that concern about how this methodology solve semantic aspect problems in different agents' system. Start from the reengineering and reusing of the multi-agent resources; conceptualize all terms and relationships; restructure resources; until formalize all terms and relationships into diagram design.

The results of ontology knowledge show how ontology solves semantic aspects by reconstructing and rearranging all information from all data sources and reforming it into meaningful information in ontology knowledge. In addition, in the results and discussion sections, we found several advantages and disadvantages of OmMAS that allow for improvements to this method in the future. Based on the various advantages and disadvantages presented in the previous section, we can improve the next stage of the methodology and try to apply it in different case studies to determine its performance.

References

1. Boes, J.; and Migeon, F. (2017). Self-organizing multi-agent systems for the control of complex systems. *Journal of Systems and Software*, 134, 12-28.
2. Bokhari, M.U.; and Ahmad, S. (2014). Detailed analysis of existing multi-agent based e-learning systems. *Proceedings of the 2014 International Conference on Computing for Sustainable Global Development* (*INDIACom*), New Delhi, India, 388-393.
3. Bokhari, M.U.; and Ahmad, S. (2014). Multi-agent based e-learning systems: A comparative study. *Proceedings of the* 2014 *International Conference on Information and Communication Technology for Competitive Strategies*, *ACM*, Udaipur, Rajasthan, India, 1-6.
4. Búrdalo, L.; Terrasa, A.; Julián, V.; and García-Fornes, A. (2018). The information flow problem in multi-agent systems. *Engineering Applications of Artificial Intelligence*, 70, 130-141.
5. Boutsioukis, G.; Partalas, I.; and Vlahavas, I. (2012). Transfer learning in multi-agent reinforcement learning domains. *Proceedings of the Recent Advances in Reinforcement Learning, 9th European Workshop, EWRL 2011*, Athens, Greece, Springer Berlin Heidelberg, Berlin, Heidelberg, 249-260.

6. Werneck, V.M.B.; Costa, R.M.E.M.; and Cysneiros, L.M. (2011). Modelling multiagent system using different methodologies, multi-agent systems - modeling, interactions, simulations and case studies *in (Alkhateeb, D. F., ed)*, InTech, 77-96.
7. Al_Asadi, H. (2012). Architectural analysis of multi-agents educational model in web-learning environments. *Journal of Emerging Trends in Computing and Information Sciences*, 3(6), 930-935.
8. Gregg, D.G. (2007). E-learning agents. *The Learning Organization*, 14(4), 300-312.
9. Liu, X.; Zhang, X.; Soh, L.-K.; Al-Jaroodi, J.; and Jiang, H. (2003). I-MINDS: an application of multiagent system intelligence to on-line education. *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, Washington, DC, USA, 4864-4871.
10. Silveira, R.A.; and Vicari, R.M. (2002). Developing distributed intelligent learning environment with JADE - Java agents for distance education framework. *Intelligent Tutoring*, 105-118.
11. Silveira, R.A.; and Vicari, R.M. (2002). Improving interactivity in e-learning systems with multi-agent architecture. *Adaptive Hypermedia and Adaptive Web-Based System*, 466-471.
12. Ying, W.; Ray, P.; and Lewis, L. (2013). A methodology for creating ontology-based multi-agent systems with an experiment in financial application development. *Proceedings of the 2013 46th Hawaii International Conference on System Sciences* (HICSS), Wailea, HI, USA, 3397-3406.
13. Yunianta, A.; Barukah, O.M.; Yusof, N.; Musdholifah, A.; Jayadiyanti, H.; Dengen, N.; Haviluddin and Othman, M.S. (2017). The ontology-based methodology phases to develop multi-agent system (OmMAS). *Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics* (*EECSI*), Yogyakarta, Indonesia, 1-6.
14. Rani, M.; Nayak, R.; and Vyas, O.P. (2015). An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. *Knowledge-Based Systems*, 90, 33-48.
15. Yunianta, A.; Yusof, N.; Bramantoro, A.; Haviluddin, H.; Othman, M.S.; and Dengen, N. (2016). Data mapping process to handle semantic data problem on student grading system. *International Journal of Advances in Intelligent Informatics*, 2(3), 157-166.
16. Yunianta, A.; Yusof, N.; Othman, M.S.; Aziz, A.; Dengen, N.; Ugiarto, M.; Haeruddin; and Angelin, J. (2014). Semantic data mapping on E-learning usage index tool to handle heterogeneity of data representation. *Jurnal Teknologi (Sciences and Engineering)*, 69(5), 1-6.
17. Tran, Q.-N.N.; and Low, G. (2008). MOBMAS: A methodology for ontology-based multi-agent systems development. *Journal of Information and Software Technology*, 50(7-8), 697-722.
18. Scott, A.D.; and Madhukar, K. (2005). Multi-agent systems engineering: an overview and case study. *Agent-Oriented Methodologies (Brian, H.-S. & Paolo, G., Eds.)*, IGI Global, Hershey, PA, USA, 317-340.

19. Padgham, L.; and Winikoff, M. (2005). Prometheus: A practical agent-oriented methodology. *Agent-Oriented Methodologies* (*Brian, H.-S. & Paolo, G., Eds.*), IGI Global, Hershey, PA, USA, 107-135.
20. Bresciani, P.; Perini, A.; Giorgini, P.; Giunchiglia, F.; and Mylopoulos, J. (2004). Tropos: an agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(3), 203-236.
21. DiLeo, J.; Jacobs, T.; and DeLoach, S. (2002). Integrating ontologies into multiagent systems engineering. *Proceedings of the 4th International Bi-Conference Workshop on Agent-Oriented Information Systems*, Bologna, Italy, 59.
22. Burrafato, P.; and Cossentino, M. (2002). Designing a multi-agent solution for a bookstore with the PASSI methodology. *Proceedings of the 4th International Bi-Conference Workshop on Agent-Oriented Information Systems*, Toronto, Canada, 57.
23. DeLoach, S.A. (1999). Multiagent systems engineering: a methodology and language for designing agent systems. *Proceedings of the Agent Oriented Information Systems '99*, 45-47.
24. Iglesias, C.A.; Garijo, M.; Centeno-Gonzalez, J.; and Velasco, J.R. (1998). Analysis and design of multiagent systems using mas-common KADS. *Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, Springer-Verlag, 313-327.
25. Teimoury, E.; Chambar, I.; Gholamian, M.R.; and Fathian, M. (2014). Designing an ontology-based multi-agent system for supply chain performance measurement using graph traversal. *International Journal of Computer Integrated Manufacturing*, 27(12), 1160-1174.
26. Gómez-Pérez, A.; and Rojas-Amaya, M. (1999). Ontological reengineering for reuse. *Knowledge Acquisition, Modelling and Management*, 139-156.
27. Dentler, K.; Cornet, R.; Teije, A.T.; and Keizer, N.D. (2011). Comparison of reasoners for large ontologies in the OWL 2 EL profile. *Semantic Web*, 2(2), 71-87.
28. Aziz, A.; Ahmed, S.; and Khan, F.I. (2019). An ontology-based methodology for hazard identification and causation analysis. *Process Safety and Environmental Protection*, 123, 87-98.
29. Zhang, H.; Guo, Y.; Li, Q.; George, T.J.; Shenkman, E.; Modave, F; and Bian, J. (2018). An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival. *BMC Medical Informatics and Decision Making*, 18(2), 129-147.
30. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; and Fernández-López, M. (2012). The NeOn methodology for ontology engineering. *Ontology Engineering in a Networked World*, 9-34.
31. Fonou-Dombeu, J.V.; and Huisman, M. (2011). Combining ontology development methodologies and semantic web platforms for e-government domain ontology development. *International Journal of Web & Semantic Technology* (*IJWesT*), 2(2), 12-25.
32. Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5-6), 907-928.

33. Miñarro-Gimenez, J.A.; Egaña Aranguren, M.; Martínez Béjar, R.; Fernández-Breis, J.T.; and Madrid, M. (2011). Semantic integration of information about orthologs and diseases: The OGO system. *Journal of Biomedical Informatics,* 44(6), 1020-1031.
34. Horrocks, I. (2007). Semantic web: the story so far. *Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, Patras, Greece, 120-125.
35. Kovalenko, O.; and Moser, T. (2011). Using explicit and machine-understandable engineering knowledge for defect detection in automation systems engineering. *Proceedings of the International Doctoral Symposium on Software Engineering and Advanced Applications* (*IDoSEAA*), Oulu, Finland, 1-5.