

An Ontology Driven, Procedural Reasoning System-Like Agent Model, For Multi-Agent Based Mobile Workforce Brokering Systems

Arash Mousavi, Md. Jan Nordin and Zulaiha Ali Othman
School of Computer Science, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Abstract: Problem statement: It has been proven that Believe, Desire, Intention (BDI) agent architecture, performs suitably in dynamic and unpredictable environments. Although BDI architecture has been formulated rigorously, implementing BDI architecture is not as straightforward as it has been promised. Nevertheless, the preeminent implementation of BDI architecture is Procedural Reasoning System (PRS) and Jadex is a Java-based platform that successfully applies PRS BDI architecture to implement BDI agents. However, Jadex utilizes XML format to represent plans, beliefs and goals which are the core of the PRS BDI agents. Although XML is a suitable format for exchanging data, it does not add any semantic to the data that it represents. **Approach:** In contrast, in our proposed Ontology driven PRS like model (O-PRS), we used Ontology with OWL format to represent Believes, Plans and Events. Our Ontology had been designed particularly for implementing BDI agents which are used in Mobile Workforce Brokering Systems (MWBS), a multi-agent system that automated the process of allocating tasks to Mobile Workforces. **Results:** This research was an endeavor to equip the PRS model with Ontology and semantic knowledge representation and reasoning capability. In this study firstly, our proposed model was formulated, then the model was been examined using a simulation tool (MWBS-SIM) which simulated MWBS in its Initialization phase. We used JADE to implement the Agents, Protégé OWL to create the Ontology and Jena as a semantic web platform by means of which MWBS is able to connect to the Ontology and apply the O-PRS model. **Conclusion:** The result of our simulation showed that O-PRS is a feasible approach. Furthermore, Ontology provided better means for knowledge sharing and navigation therefore, multi-agent systems constructed using O-PRS, were more generic and more adaptable in comparison with those that use XML.

Key words: Procedural reasoning system, mobile workforce brokering system, ontology, multi-agent systems

INTRODUCTION

Implementing intelligent systems that can cope with dynamic and uncertain environments is one of the fundamental challenges of modern Artificial Intelligence. As a new approach to AI, Agent and Multi-agent technologies have promised to address this kind of challenges by introducing agent architectures which utilize high-level and human-centric reasoning mechanisms. BDI agent is one of the aforementioned architecture, which uses Believe, Desire and Intention that are used correspondingly to symbolize and model an agent's information state (what agent knows about itself and its environment), motivational state (what agent tries to achieve) and deliberative state (a plan to achieve agent's desired state of affair) (Rao and Georgeff, 1995; Corchado and Laza, 2003).

BDI agent architecture: BDI agents exercise a particular mechanism called practical reasoning to exhibit behaviors in their environments (Rao and Georgeff, 1995; Bratman *et al.*, 1988). Unlike theoretical reasoning, practical reasoning is reasoning to find out the most suitable actions to be performed in a particular state of the environment. This process includes two activities namely; deliberation (making decision on what state of affairs an agent wants to achieve) and means-end reasoning (to find out how to achieve that state of affair) (Wooldridge, 2002).

Similar to any other technologies, there are some advantages and disadvantages that emerge in BDI architecture too. BDI agent architecture is human-centric, easy to understand and it has been rigorously formulized (Tweeddale *et al.*, 2007). However, there is lack of an efficient implementation mechanism

Corresponding Author: Arash Mousavi, School of Computer Science, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

(Corchado and Laza, 2003) in this model. The reason according to Rao and Georgeff (1995) is that; there is a big gap between formalization logic of the BDI agents and the practical systems in which they have to act.

Procedural Reasoning System (PRS): PRS is one of the first and the most robust agent architecture that has been used to implement BDI agents (Wooldridge, 2002). PRS architecture has been tested in several different real-world multi-agent applications such as OASIS, SWARM and SPOC, and proved its power in performing the complex tasks in dynamic and uncertain environments accurately.

In a PRS-like system, an agent is equipped with a library of precompiled Plans (Wooldridge, 2002). The main task of a PRS like system then, is to select and execute plans which are essential to achieve a goal or to respond to an event. Furthermore, PRS reduces the abstract notions of desires and intentions to the more concrete concepts of goals and plans (Pokahr *et al.*, 2003).

Jadex BDI model: A successful implementation example of PRS-like system is Jadex, an add-on to Java Agent Development Platform (JADE) which utilizes PRS architecture (Pokahr *et al.*, 2003). Jadex and its BDI reasoning engine can be used to develop BDI agents capable of practical reasoning. The BDI agent model used in Jadex is a hybrid model, including structural and behavioral parts. The structural parts of Jadex BDI agent consisting of beliefs, goals and plan types are represented as an XML file called Agent Definition File (ADF). For behavioral part however, agent designers have to write codes within a class structure which is called Plan. Jadex reasoning engine would be triggered upon occurrence of an event that in turn stimulates the reasoning engine into finding and executing the appropriate plan(s) (Bratman *et al.*, 1988).

An ontology driven approach to PRS: Although XML is a suitable format for exchanging data, it only represents the syntax of the domain data but not the semantic of it. In contrast, the Web Ontology Language (OWL) is capable of adding semantic to data (Bohring and Auer, 2005). Furthermore, using an appropriate semantic web framework, knowledgebase with OWL format can be queried and reasoned over. These capabilities are convincing reasons to nominate OWL to propose a PRS-like architecture for agents that have to deal with semantic knowledgebase and ontology (semantic agents).

In our attempt to develop an efficient MWBS, we proposed O-PRS agent model, capable of manipulating

semantic knowledgebase. In this model, we adapted PRS and modified Jadex model by replacing its XML based ADF with an Ontology which includes the Believes, Plans and Events. Furthermore, in O-PRS, Taxonomy of the domain and the mental states of the agents are integrated in a unified Ontology therefore, the process of deliberation can be implemented via simple SPARQL query statements which make this process easy to implement and execute. Moreover different agents do not have to use different ADF files or different knowledge formats.

MATERIALS AND METHODS

Mobile Workforce Brokering System (MWBS): The rapid advancement in mobile communication caused a huge increment in mobile workforce population of the world, which consequently created a high demand for automated Mobile Workforce Management (MWM) systems (Chiu *et al.*, 2005). An important part of every MWM is Resource Allocation which consists of two sub tasks namely, Matchmaking and Brokering. The former refers to finding the type of resources that can handle a particular task and the latter refers to assigning a task to the most suitable and available resource of that type, which happened to be more challenging too.

In mobile environments, Resource Brokering process has to address two types of risks; Human Recourse risks (unpredictable unavailability of the workforces) and Environmental risks (e.g., disconnection). That is because mobile workforces are combination of human and mobile devices (like PDA) and they are not passive and also disconnection may happen at any time in a mobile network (Mousavi and Nordin, 2007).

We believe that using a multi-agent approach and an Ontology which represents Domain Knowledge, History of the system and organizational Policy, the above mentioned risks can be addressed. Therefore, we implemented a Multi-agent based MWBS, which automates the process of task allocation to Mobile Workforces and addresses the aforementioned risks. Figure 1 demonstrates the architecture of this system.

The components of MWBS: MWBS, as the architecture illustrates, is composed of several agents who coordinate in order to automate the task allocation process for mobile workforces, who are connected to the system via mobile network and their PDAs. MWBS, in its life time undergoes two phases; Initialization and Runtime. At initialization phase, the main goal is to figure out the realistic monthly capacity of the system, which is the sum of monthly capacities of entire individual workforces, which means number of days that a mobile workforce is able to work per month.

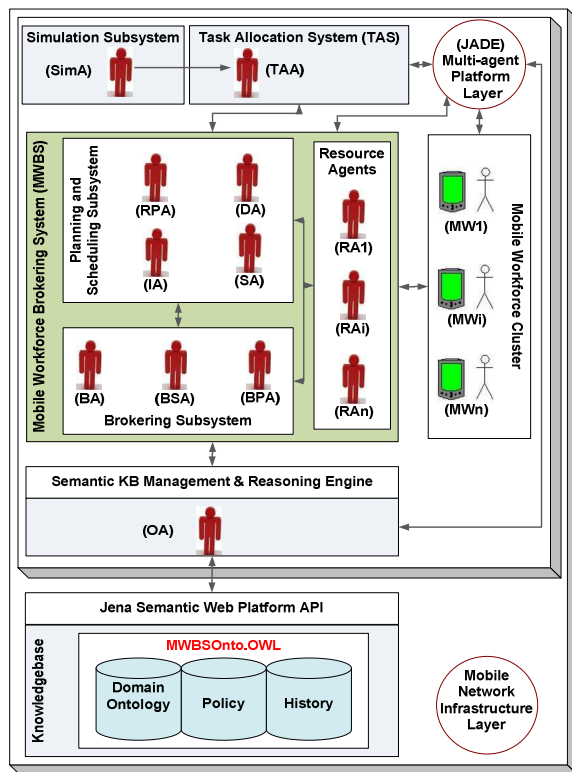


Fig. 1: The architecture of MWBS

On the other hand, the main goal of the Runtime phase is to assign tasks to appropriate workforces in a way that, the monthly capacity, which has been claimed during the initialization phase, will be accomplished accurately and fairly

Justification of using ontology driven approach: As it has been described in (Hadzic *et al.*, 2009) the volume of the available information is increasing rapidly, which in turn makes it difficult for human to browse through or manipulate them. Apparently, the reason is that information is currently represented in a semantically poor format, which means it is easily understood by people but not by machines. In contrast, semantic technology empowers the computer systems by enabling them to represent the information in semantically rich format, which means easily understood by computers. Utilizing semantic technology, machines will be able to extract meaning from the Information and to process them in an automatic fashion, with less human involvement.

Likewise, Intelligent Software Agents benefit from semantically represented knowledge too. According to Berners-Lee *et al.* (2001), the actual influence of semantic technology will be uncovered when the people

are able to develop intelligent agents capable of acquiring knowledge from different sources, manipulating them and sharing the results amongst them. Utilizing the power underpinning semantic technology, agents are able to perform the entire aforementioned activities automatically.

In addition, according to (Gruber, 1993), an Ontology is an explicit specification of a conceptualization. This definition implies that ontology is an abstract model of the world which includes concepts, their properties and relationships which relate them, described in an unambiguous language. Therefore, Ontology is a sufficient way for representing domain knowledge, which can be shared by the agents who are situated in the domain environment. Thus, semantic web provides more flexibility for agents by enabling them to share Ontologies, which in turn provide the agents with the vocabulary that they need to communicate (Hadzic *et al.*, 2009; Berners-Lee *et al.*, 2001). Furthermore, agents can extend domain knowledge and navigate through it using automated reasoning and logical approach.

In MWBS, since the entire domain knowledge is represented semantically using OWL ontology, applying the same approach to represent the knowledge that BDI agents need for their practical reasoning process, provides consistency for the system as a whole. Moreover, MWBS can exhibit more generality by means of Ontology as if the domain knowledge, policy and the plan are changed, it will affect only the system in its knowledge level not in process level. Therefore, by updating or totally replacing the ontology with a new one, system will demonstrate new set of behaviors, which makes the MWBS a generic system capable of adapting to different environments. Therefore, consistency, adaptability and generality are three main characteristics that using ontological approach to PRS like BDI model will provide specifically to agents employed in MWBS. Gaining these benefits thus, justifies conducting this research.

Jadex model: According to (Bellifemine *et al.*, 2007) the main components of Jadex model are Believe, Goal, Plan and an practical reasoning interpreter that its responsibility is to select a proper set of goals (goal deliberation phase), upon receiving a message or occurrence of an event. Next, the interpreter should find a proper plan that its execution achieves agent's goals that have been identified during goal deliberation phase (Means-end reasoning) (Bratman *et al.*, 1988; Pokahr *et al.*, 2005). In addition, in this model, a set of plans are assigned to each agent. Each plan is a set of actions that an agent can perform and are selected based

on the capabilities of the agent as well as the objective of each plan. The entire list of goals, believes and plans of an agent is represented in a file called ADF, while the actual plans (the code) are represented in a Plan class file which is written in Java. By finding an appropriate plan for an agent in a specific situation, Jadex reasoning engine will be able to execute the code assigned to the plan and the result will be the fulfillment of the goal.

The first difference between Jadex and O-PRS: The first major difference between Jadex and O-PRS however, is that we do not directly represent the concept of goal in O-PRS. Instead, each agent may have one or more Believe (s) which may be corresponded to certain events and plans. Therefore, an Event might be assigned to a Believe and if so, an appropriate Plan is assigned to the same Believe as well, which means if that Event occurs for the agent, the corresponding Plan for the same Believe will be fired. This concept is depicted in Fig. 2 which shows that if, for instance, Event1 is fired for Agent, it implies that current desire of the Agent is Believe_1 and therefore, Plan 1 should be selected and executed in order to fulfill this desire.

Upon occurrence of an Event for a particular agent, a Plan Selection Function will be fired that searches within the agent's Believe set to find out a particular Believe to which that Event is assigned and if it finds any, it will return the corresponding Plan which is assigned to that Believe. Having the Plan signature (its name), then the agent can run the appropriate code which is written to fulfill the Plan. With this explanation, we modeled the O-PRS in the following lines.

Formal model of O-PRS: This part of this study illustrates the formal model of O-PRS. We preliminarily use the notation which is used in (Wooldridge, 2002) and then extend the model by adding more functions which are required, specifically for O-PRS.

Definition 1: Let B, P, A, E and AC be the sets of entities which exist in the model. We define each set as follow:

- B = {b | b is a valid Believe in the system}
- P = {p | p is a valid Plan in the system}
- A = {a | a is a valid Agent in the system}
- E = {e | e is a valid Event in the system}
- AC = {α | α is a valid Action in the system}

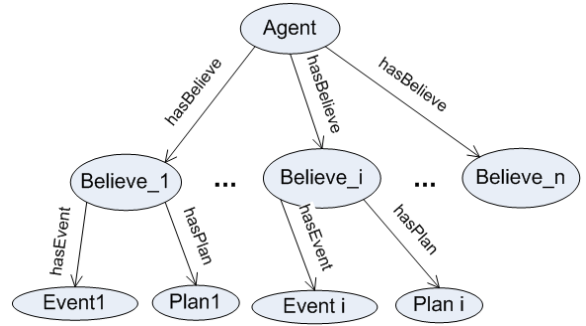


Fig. 2: Representing agent, believe and event in O-PRS ontological model

Definition 2: The Basic (Atomic) Functions of the Model are defined as follow:

- hasBeleive(a,b): Evaluates if agent a, has Believe b or not.
- hasPlan(b,p): Evaluates if a Plan p is assigned to Believe b or not.
- matchBelieve(b,e): Upon occurrence of an Event e, evaluates if e is assigned to Believe b or not.

Definition 3: The Believe function of agent $a \in A$; Bel(a) produces the Believes set of agent a. The Signature of the Bel(a) of agent $a \in A$ is:

$$\text{Bel}(a): B \rightarrow B$$

$$\text{Bel}(a) = \{b \mid b \in B \wedge \text{hasBelieve}(a,b)\}$$

Definition 4: The BelieveOfPlan is a function that generates a set of all believes like b of agent $a \in A$, which has the plan $p \in P$ assigned to it:

$$\text{BelieveofPlan}(a,p) = \{b \mid b \in \text{Bel}(a) \wedge \text{hasplan}(b,p)\}$$

Definition 5: The Intention function Int(a) of agent $a \in A$ generates a set of all plans that satisfies at least one believe $b \in \text{Bel}(a)$. The Signature of Int(a) of $a \in A$ is:

$$\text{Int}(a): P \rightarrow P$$

$$\text{Int}(a) = \{p \mid p \in P \wedge (\text{BelieveOfPlan}(a,p) \neq \emptyset)\}$$

Definition 6: The Options function Option(a) is a function which generates possible Desire set for agent $a \in A$. The signature of Options(a) of $a \in A$ is:

$$\text{Options}(a): \text{Bel}(a) \times \text{Int}(a) \rightarrow \text{Des}(a)$$

$$\text{Des}(a) = \{(b,p) \mid b \in \text{Bel}(a) \wedge p \in \text{Int}(a) \wedge \text{hasPlan}(b,p)\}$$

Definition 7: The Plan Selection is a function which after the occurrence of an event $e \in E$, will select a plan for Agent $a \in A$ to which event has occurred. The signature of $\text{planSelection}(a,e)$ of $a \in A$ and $e \in E$ is:

$$\begin{aligned} \text{planSelection}(a,e) &: \text{Des}(a) \rightarrow \text{Int}(a) \\ \text{planSelection}(a,e) &= \{p \mid p \in \text{Int}(a) \wedge \\ &(\forall(b,p) \in \text{Des}(a) \rightarrow \text{matchBelieve}(b,e))\} \end{aligned}$$

In present model either no plan or, only and only one plan will be assigned to each believe, therefore $\text{planSelection}(a,e)$ either select only one plan, or nothing. Following lines describe, how this model can be implemented using an ontology driven approach.

Representing practical reasoning in ontology:

The second difference between Jadex and O-PRS: The second major difference between O-PRS and Jadex is that the latter represents plans and goals in an XML format, while the former utilizes ontology to relate these entities.

The vocabulary of OWL ontology: Figure 3 (adapted from (Ding *et al.*, 2007)) illustrates the Semantic Web layer cake. In Fig. 3, it is depicted that, the semantic is enabled by a pile of evolving languages such as Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL) (Ding *et al.*, 2007) amongst them OWL provides mechanisms and constructs to create actual semantic knowledge bases. In an ontology that has been represented in OWL, the concepts of a domain are represented as Classes and the Individuals represent the instances of a particular class. Classes and their relationships, shape the Taxonomy of the domain (Hebeler *et al.*, 2009) which is known as T-Box. On the other hand, individuals which are related to one another via relationships called ObjectProperties, create assertions which are called A-Box or assertion box. The attributes of an individual is represented by DataProperties that can be of any valid types such as Integer and String, and relates the individual to a value in the valid range of its type. Using vocabulary of OWL, we can represent the domain knowledge of any domain.

Components of MWBS domain ontology: In the MWBS domain, the most fundamental class is Agent class which has subclasses such as; Task Allocation Agent, Dealer Agent or Supervisor Agent. Furthermore O-PRS model includes other classes such as Believe, Plan and Event. Event has a sub class called Message that represents the messages which are exchanged between agents. As it is illustrated in Figure 4,

hasBeleive Object Property relates the individuals of class Agent and the individuals of class Believe. Since each individual of Believe may have some event, hasEvent objectProperty relates Believe and Event classes. While an individual of class Believe may have a plan too, therefore class Believe and Plan are connected via hasPlan ObjectProperty. Moreover, there are three DataProperties being used in this ontology: hasSignature which assigns a unique name of the string data type to each plan, hasPerformative and hasContents both of string data type which add some communicational attributes to the individuals of the Message class. The OWL Ontology used to represent knowledge for MWBS is called MWBSOnto.owl.

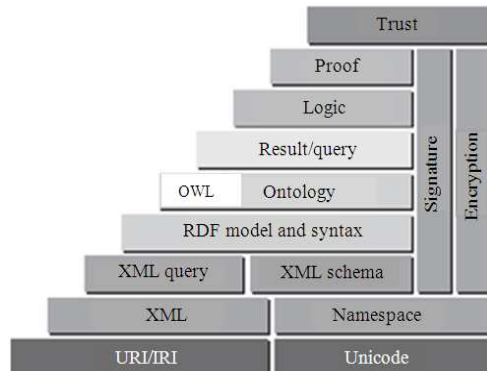


Fig. 3: Semantic web layered cake (Ding *et al.*, 2007)

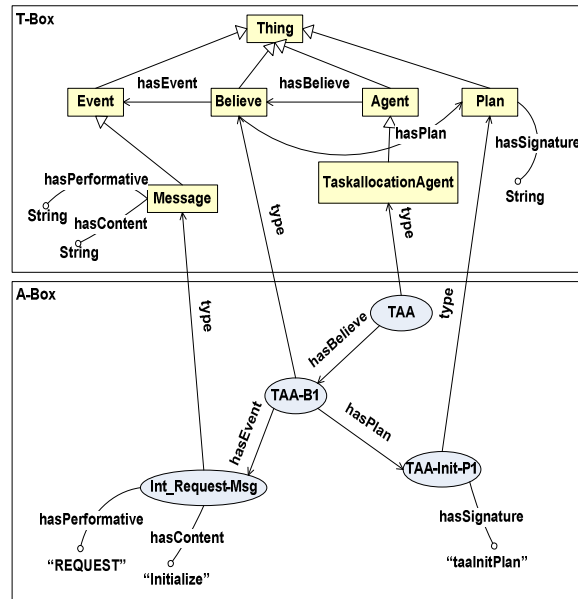


Fig. 4: Semantic representation of the initialization plan for a task allocation agent in MWBSOnto.owl

Plan finding in O-PRS: According to Foundation of Intelligent Physical Agents (FIPA) standard, every message has a Performative that identifies the message type, and Content. The combination of these two sufficiently identifies a message, which should be interpreted by the agent that receives it. In O-PRS, in order to interpret a message, agents can simply use combination of <performative, content, receiver agent's name> to compose and apply an SPARQL Query statement to firstly find the corresponding Message in the ontology and secondly to find a Believe of the receiver agent to which the message belongs, and finally find the Plan which is assigned to the found Believe and return its signature.

For instance, as Fig. 4 demonstrates, TAA is an instance of TaskAllocationAgent class. TAA has a Believe called TAA-B1 to which an instance of Message class has been assigned called; Int_Request_Msg. The performative of this message is the string "REQUEST" and the content of this message is the string "Initialize". An Individual of Plan class also is assigned to the message, called; TAA-Init-P1 with plan signature of string type "taaInitPlan". This portion of the MWBSOnto implies that; If Agent TAA receives a message which is a "REQUEST" and its content is "Initialize", then it has to execute the plan "taaInitPlan". The first part of this scenario that is to find the message that asks for an initialization to take place is called deliberation and the second part is called means-end reasoning, which both can be achieved via running the SPARQL Query shown in Figure 5. This

query statement returns the signature of the TAA-Init-P1 plan, which is the string value; "taaInitPlan".

Developing and examining the ontology: The Ontology that O-PRS needs is created in Protégé OWL. Figure 6 illustrates a screenshot of the MWBSOnto.owl in Protégé.

The Agents, however, are created in Java Agent Development Platform (JADE). In addition Jena Platform has been used to make the connection between agents and ontology and execute the SPARQL query.

```

PREFIX MWBSOnto: <http://MWBSOnto.owl#>
SELECT ?believe ?message ?plan ?planSignature
WHERE {
  MWBSOnto:TAA MWBSOnto:hasBelieve ?believe.
  ?believe MWBSOnto:hasEvent ?message.
  ?message MWBSOnto:hasPerformative "Request";
  MWBSOnto:hasContent "Initialize".
  ?believe MWBSOnto:hasPlan ?plan.
  ?plan MWBSOnto:hasSignature ?planSignature
}
    
```

Fig. 5: SPARQL Query to find a plan signature in MWBSOnto.owl ontology

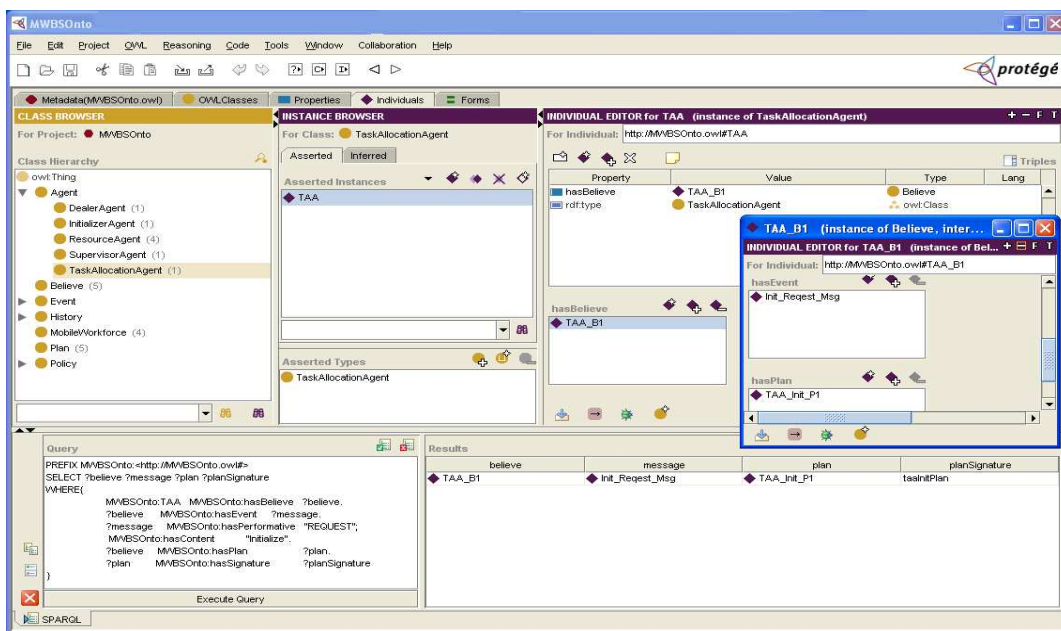


Fig. 6: Representation of MWBSOnto.owl ontology and query execution in protégé OWL

Plan execution (means-end reasoning): In order to examine the O-PRS model, we have implemented a simulation tool called MWBS-SIM with the same architecture as depicted in Fig. 1. In this architecture users start the simulation and communicate with the system via Simulation Agent (SA). MWBS-SIM also utilizes an Ontology Agent (OA) via which, other agents can connect to the ontology. OA, using Jena classes is able to connect to OWL ontology, search and navigate it and execute the SPARQL Queries over it.

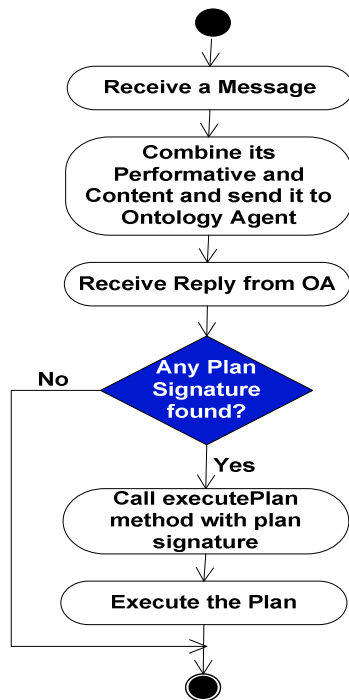


Fig. 7: Plan selection and execution flowchart

```

public class Plan {
...
public void executePlan(TaskAllocationAgent a, String planSignature) {
    if (planSignature.equals("taaInitPlan")) {
        taaInitPlan(a); } ...
void taaInitPlan(final TaskAllocationAgent a) {
    /* 1- Find an Instance of Dealer Agent
    /* 2- Send a request message to it,
        Performative= "Request" Content = "Initialize" */
    a.addBehaviour(new OneShotBehaviour() {
    public void action() {
        ACLMessage init_msg =
            new ACLMessage(ACLMessage.REQUEST);
        init_msg.addReceiver(newAID("DA",
            AID.ISLOCALNAME));
        init_msg.setLanguage("English");
        init_msg.setContent("Initialize");
        a.send(init_msg);
    } }); }...}
    
```

Fig. 8: Partial view of plan class for TAA

In this architecture an agent (we call it requester agent) upon receiving a message, composes a Request message with the content that is the combination of performative and the content of the received message and send it to OA to find if any plan is assigned to that message (deliberation) by running an appropriate SPARQL Query over the MWBSOnto.owl. If OA could not find any plan it will send a null reply message back to the requester agent, otherwise, it will send its signature back to the requester agent (means-end reasoning). Next, the requester agent will call the executePlan method of the Plan class with the plan signature that OA found which in turn executes the plan.

The state diagram shown in Fig. 7 summarizes this process. Figure 8 demonstrates a partial view of the Plan class (written in Java) which is the implementation of executePlan method for TAA. If the plan signature is "taaInitPlan" then the function taaInitPlan will be run, which in this case, sends a message to Dealer Agent (DA) with REQUEST performative and Initialize content.

RESULTS

Table 1 lists the agents who are involved in Initialization phase of MWBS life time. Using O-PRS, these agents can find each other and send appropriate messages to appropriate agents in order to initialize the system. The main goal of this phase as mentioned earlier is to prepare a Realistic Monthly Capacity (RMC) for the system.

Simulation starts when SimA sends a REQUEST message with "Initialize" content to TAA. Using the mechanism described earlier, TAA finds its plan and executes it which as result, sends a request message to DA, DA sends the same message to SA which then sends the same message to IA. At this point, IA will find all RAs and then sends initialization request to all of them.

As depicted in Fig. 1, each RA represents a particular mobile workforce in the system. RAs then ask their corresponding mobile workforces to submit a proposal which includes the monthly capacity and a-schedule of their choice. Upon receiving a proposal, each RA evaluates the validation of the proposal by applying the policy which are enforced by the authorities and represented semantically in MWBSOnto.owl Ontology (proposal evaluation is beyond the content of this study).

Table 1: MWBS's subsystems and agents

Corresponding agent	Functionality
Planning-Scheduling Subsystem (PSS)	
Supervisor Agent (SA)	Supervise agents of PSS
Dealer Agent (DA)	Make deal with TAS
Initializer Agent (IA)	Create initial plan
Other Functionalities and Agents	
TaskAllocationAgent (TAA)	Deal with MWBS
Resource Agent (RA)	Represent mobile workforce
Simulation Agent (SimA)	Starts simulation
Ontology Agent (OA)	Knowledgebase operations

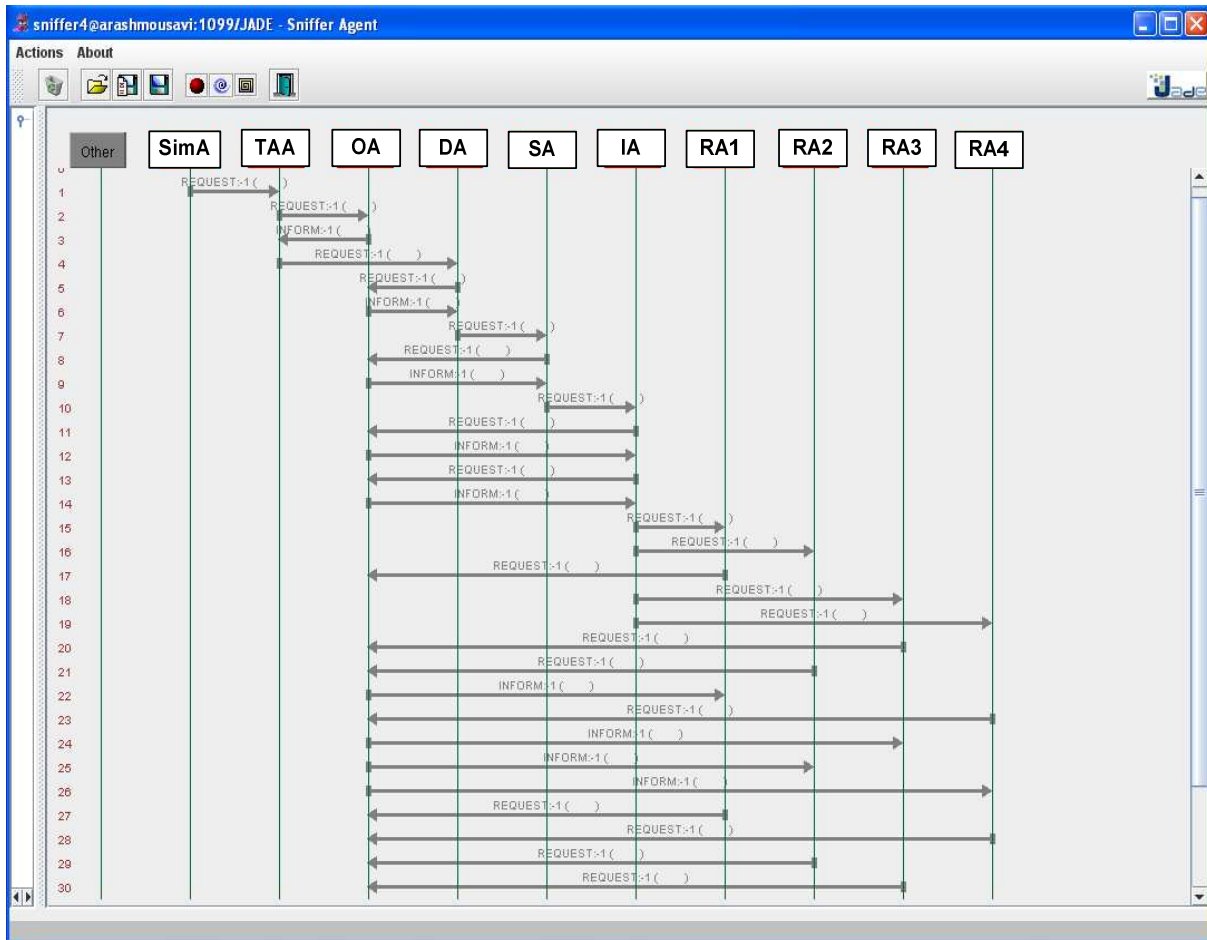


Fig. 9: JADE sniffer, illustrates the agents communication using O-PRS architecture

The final result however is a monthly proposal that will be passed to DA which in turn will be passed to TAA. Figure 9 illustrates a screenshot of JADE's Sniffer Agent. It shows how the agents involved in this scenario are communicating to initialize the system, using O-PRS BDI architecture.

DISCUSSION

In this study we have proposed an Ontology Driven PRS like BDI architecture called O-PRS. Our proposed model utilizes semantic knowledge representation and reasoning and ontological approach to represent the basic knowledge that PRS-like agents need to act, which are Believe, Event and Plan.

In this study we have shortly described the PRS model used in Jadex and have gone through exploring the advantages of using OWL Ontology to represent the basic components of this model in comparison with

XML, which is used in Jadex. Our discussion has been followed by developing a formal model for O-PRS. Furthermore we have shown the internal architecture of MWBSOnto.owl, the Ontology which has been utilized by O-PRS and the mechanism using which O-PRS agents are able to brows through the Ontology in their deliberation and means-end reasoning process.

We have also portrayed MWBS, the system in which O-PRS has been implemented and examined. In addition, via a simulation, using our simulation tool MWBS-SIM, it has been shown that O-PRS is applicable and it is suitable for being used in mobile environments.

CONCLUSION

In conclusion, the result of this research proves that; ontology and semantic knowledge representation and reasoning techniques can be used as effective tools

to empower agents who are situated in dynamic environments in performing their tasks, as well as in communicating with each other, in a robust and acceptable manner. Multi-agent systems, which are implemented using this approach, are more generic, adaptable and consistent in comparison with other approaches.

REFERENCES

- Bellifemine, F., G. Caire and D. Greenwood, 2007. Developing Multi-Agent Systems with JADE. 1st Edn., John Wiley and Sons Ltd., ISBN: 978-0-470-05747-6, pp: 254.
- Berners-Lee, T., J. Hendler and O. Lassila, 2001. The semantic web. *Sci. Am.*, 284: 34-43. <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>
- Bohring, H. and S. Auer, 2005. Mapping XML to OWL ontologies. *Leipziger Informatik-Tage. LNI*, 72: 147-156. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.8897>
- Bratman, M.E., D.J. Israel and M.E. Polak, 1988. Plans and resource-bounded in practical reasoning. *Comput. Intell.*, 4: 349-355. DOI: 10.1111/j.1467-8640.1988.tb00284.x
- Chiu, K.W., S.C. Cheung and H. Leong, 2005. A multi-agent infrastructure for mobile workforce management in a service oriented enterprise. *Proceedings of the IEEE 38th Hawaii International Conference on System Sciences (HICSS'05)*, IEEE Computer Society, Track 3, pp: 85-94. DOI: 10.1109/HICSS.2005.28
- Corchado, J.M. and R. Laza, 2003. Constructing Deliberative agents with case-based reasoning technology. *Int. J. Intell. Syst.*, 18: 1227-1241. DOI: 10.1002/int.10138
- Ding, L., P. Kolari, Z. Ding, S. Avancha and T. Finin *et al.*, 2007. Using Ontologies in the Semantic Web: A Survey, *Integrated Series in Information Systems-Ontologies*. Springer, US., pp: 79-113. ISBN: 978-0-387-37019-4.
- Gruber, T., 1993. A Translation approach to portable ontology specification. *Knowl. Acquisit.*, 5:199-220. DOI: 10.1006/knac.1993.1008
- Hadzic, M., P. Wongthongtham, T. Dillon and E. Chang, 2009. *Ontology-Based Multi-Agent Systems. Studies in Computational Intelligence. Vol. 219*, Springer-Verlag Berlin Heidelberg, ISBN: 978-3-642-01903-6, pp: 3.
- Hebeler, J., M. Fisher, R. Blace and A. Perez-Lopez, 2009. *Semantic Web Programming. 1st Edn.*, Wiley Publishing, Inc., USA, ISBN: 978-0-470-41801-7, pp: 99.
- Mousavi, A. and M.J. Nordin, 2007. An architectural model for a multi-agent mobile workforce brokerage system based on CBR-BDI agent architecture and active shared-data space coordination model. *Proceeding of the IEEE International Conference on Electrical Engineering and Informatics Institute Technology Bandung*, June 17-19, Indonesia, pp: 294-297. <http://repository.gunadarma.ac.id:8000/667/1/B-42.pdf>
- Pokahr, A., L. Braubach and W. Lamersdorf, 2005. Jadex: A BDI reasoning engine. *Multi-Agent Programm.*, 15: 149-174. DOI: 10.1007/b137449
- Pokahr, A., L. Braubach, W. Lamersdorf, 2003. Jadex: Implementing a BDI-Infrastructure for JADE Agents. *EXP-In Search of Innovation. Special Iss. JADE*, 3: 76-85. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.3969>
- Rao, A.S. and M. Georgeff, 1995. BDI Agents from theory to practice. *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, USA., pp: 312-319.
- Tweeddale, J., N. Ichalkaranjeb, C. Sioutisb, B. Jarvisb, A. Consolib and G. Phillips-Wrenc, 2007. Innovations in multi-agent systems. *J. Network Comput. Appl.*, 30: 1089-1115. DOI: 10.1016/j.jnca.2006.04.005
- Wooldridge, M., 2002. *An Introduction to Multiagent Systems. 1st Edn.*, John Wiley and Sons Ltd., ISBN: 0-471-49691 X, pp: 65.