International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 8, No.1 (Jan-2019)

http://dx.doi.org/10.12785/ijcds/080101

An Agents' Model Using Ontologies and Web Services for Creating and Managing Virtual Enterprises

Mahmoud Brahimi[1]

[1] *Department of Computer Science, Mohamed BOUDIAF University, M'Sila, Algeria*

Received 20 Sep. 2018, Revised 30 Nov. 2018, Accepted 9 Dec. 2018, Published 1 Jan. 2019

Abstract: Currently, the economic competition has become very tough where successful companies find themselves obliged to develop their organizations and diversify their work strategies. Thereby, virtual enterprises concept imposes itself and overcomes geographical constraints by allowing companies to cooperate and address their shortcomings through exchanging their services and skills. The present paper proposes a virtual enterprise model based on the use of agents endorsed by the use of ontologies and web services. The coupling of these concepts allows reducing the high costs associated with the use of technology and improves flexibility and intelligence both in the negotiation process through the selection of the best partners, and in the managing process by ensuring the effective cooperation between partners.

Keywords: Virtual Enterprises, Agents, Ontologies, Web Services, Negotiation, FIPA-ACL

1. Introduction

Technology has, in fact, greatly accelerated the effects of globalization and pushed enterprises, in a competitive challenging context, to adapt their way of organizing with the newly existing changes, and to overcome the spatio-temporal variables by reorganizing themselves constantly in order to respond "in real time" to the needs expressed by the socio-economic world. In addition, the rise and commercialization of the Internet and the evolving of information and communication technologies (ICT) are making organizations' business environments increasingly more international, contributing by that to the creation of what is called electronic business.

Nowadays, enterprises need to store and manage a mass of important information, where managers must seek a great flexibility in using their data and making them faster and more economic. In this regard, electronic business has emerged especially to exploit all kinds of communication media to overcome management difficulties, and to improve services offered to customers and speed up work procedures. This way of working has indeed led to the appearance of virtual activities in the international economic world and consequently to the emergence of virtual enterprises. It is from this perspective of omnipresence and competitiveness that the concept of virtual enterprise appears.

Several definitions have been given to this concept but the most general and operational one is that of Cruz-Cunha and Puntik [1]. According to them, a virtual enterprise is a dynamically reconfigurable global networked organization, networked enterprise, or network of enterprises, sharing information and/or knowledge, skills, core competencies, market and other resources and processes, configured (or constituted) as a temporary alliance (or network) to meet a (fast changing) market window of opportunity, presenting as main characteristics agility, virtuality, distributivity, and integrability.

Nevertheless one of the main underlying trends in research and development in this area is the focus on models, protocols, and mechanisms to support the collaboration of pre-existing entities in distributed environments, be it among organizations, among people, or among people and organizations [2]. Technically speaking, virtual enterprises require strongly a good cooperation process in order to achieve their goals. In this regard, the need for intelligent adaptive systems for managing and controlling this process is highly felt. Thereby, the multi-agent system approach offers favorable prospects for this preoccupation.

Moreover, it is crucial to select the most appropriate partners when forming a virtual enterprise and ensuring communication and cooperation between them effectively. It is in this sense that the concept of

E-mail:mahmoud.brahimi@univ-msila.dz

ontologies takes its importance by facilitating the choice of partners and enhancing communication and effective co-operation between them, especially when these are founded around multi-agent systems [3]. In addition, the high rate of interactions between virtual enterprise partners and the need to reduce response times, require a more loosely-coupled, peer-to-peer, dynamic, fast, flexible and proactive mode of interaction. Therefore, Web services have the potential to guarantee these needs by enabling interoperability between partners through different software running on different platforms based on open standards and protocols, irrespective of the languages that have been used to implement the Web Services and the platforms where the Web Services are executed. Web service opportunities led to the appearance of Web Service-based Virtual Enterprise concept (WSVE) which refers to the virtual enterprise constructed with Web service as the supporting technology [4].

In this research paper, we propose a model for creating and managing virtual enterprises that is based on the coupling of agents, ontologies and Web services, where the agent's paradigm enhances the intelligence both in the cooperation and the negotiation process. Besides, ontologies and Web services allow the correct choice of partners and ensure an effective interaction and exchange services among them.

The rest of this paper is subdivided into three main sections. The first section entails a related literature in which the contexts of virtual enterprises and multi agent systems are introduced. The second section introduces a detailed description of the proposed model, and finally the last section within which the conclusion and some suggestions for future work are provided.

2. RELATED LITERATURE

Over this section some concepts related to virtual enterprises will be defined, and accompanied with an overview of their most relevant aspect. Subsequently the contributions of multi-agent systems in the evolution and the maturation of this field will be highlighted.

A. Overview of Virtual Enterprise

Virtual enterprises are endowed with evolutionary organizational models that are very fluid and flexible. These models can be described as a group of collaborators who unite quickly to take advantage of a specific opportunity in the market. Virtual enterprises have various characteristics, according to Wassenaar [4] these characteristics can be divided into intra-organizational and inter-organizational characteristics. The former deal with the creation of Virtual Enterprises within organizations, and the latter consider the formation of Virtual Enterprises as a network organization. In this context, the most active research field is the one that deals with the inter-organizational characteristics because they evolve hard challenges and they must ensure the following features [5]:

- Virtuality; virtual enterprises are based on the use of ICT. This latter enables geographically dispersed enterprises to join each other in a virtual enterprise to achieve business goal.
- Dynamics; a virtual enterprise is highly dynamic and its life cycles can be very short.
- Flexibility; virtual enterprises have strategic objectives to maximise flexibility and adaptability to environmental changes;
- Autonomy; to respond to fast changing environment and to enable flexibility.
- Heterogeneity and immobility; these qualities constitute the cornerstone of virtual enterprises' success because they reflect diversity of firms which own different information, knowledge, and skills. These qualities ensure important new market opportunities.

In addition to these challenges, a hard managerial work must be added when establishing virtual enterprises. In this vein, we are talking about virtual enterprises management which fits the field of project management. This latter is the application of knowledge, skills, tools and techniques in the project activities to meet its requirements [6].

The established success of a virtual enterprise relies strongly on the good management of its lifecycle which can be defined as a period between its creation and integration until its dissolution, comprehending its operation and including its reconfigurations [7].

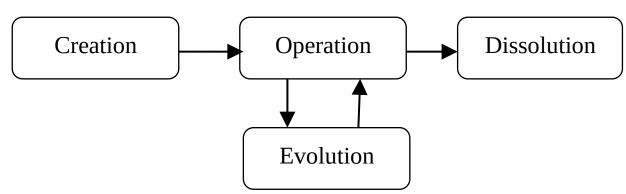

Figure 1. Virtual enterprise lifecycle [8]

The lifecycle of a virtual enterprise consists mainly of the following phases: creation, operation / evolution and dissolution (see figure 1). These phases encompass several activities that should be ensured by high performance. The good coordination and cooperation between all the partners are indispensable. The virtual enterprise activities intervening in its lifecycle are summarized by [8] and [9] below:

1) The creation phase

It allows creating / configuring a virtual enterprise. Once a business opportunity is identified, there is a need to quickly plan a suitable virtual business, identify partners and establish cooperation agreements. The creation phase generally aims to form a group of companies to carry out a business process, which must be carried out by a set of companies. The main activities required during this phase are:

- Search for partners: this activity consists of publishing an announcement of the specifications of the enterprise's needs, including all the information (technical aspects, quality, etc.), as well as the research and selection of partners who meet the requirements mentioned in the announcement.
- Negotiation: The negotiation process can be used to select partners, with a given duration, given quality and a given price with clear and well defined objectives.
- The establishment of legal agreements: The concept of virtual enterprise does not replace the need to legalize the agreements reached. As a result, the contract associated with the offer must be duly signed and legalized.

2) Operation / Evolution phase

During this phase, business members execute their business processes; consequently they achieve their common goal. The main activities required during this phase include business process management to ensure the achievement of the goal of the virtual enterprise, the work supervision to deal with disruptions, and finally the inclusion or removal of members from the virtual enterprise.

3) Dissolution phase

This phase begins when the virtual enterprise finishes the execution of its business process. The objective here is to dismantle the created virtual enterprise.

B. Multi agent systems in Virtual enterprises modeling

From all what has been stated above, it can be clearly deduced that most of the characteristics in the virtual enterprises domain make it a suitable application area for multi agent systems. The agent metaphor can be viewed as an encapsulated problem solving entity which exhibits the properties of autonomy, social ability, responsiveness and proactiveness [10].

Thanks to agent's characteristics, the agent became recently popular with distributed, large-scale, and dynamic applications. It can play many roles in the advanced information systems and e-business areas (see figure 2).

Basically virtual enterprises are composed of distributed, heterogeneous and autonomous components which need to establish between them some features based essentially on negotiation, cooperation, coordination, decision-making and information exchange. These situations are easily mapped into multi agent systems.

Due to these positive arguments in favour of the use of multi agent systems, many works are proposed using agent paradigm for creating and managing virtual enterprises. The search and selection of partners are among the most problems dealt with by agent paradigm. This process comes first in the creation phase and the choice of good suppliers constitutes the key success for the rest of the virtual enterprises. Yang et al. have proposed a multi-agent based partner selection platform in order to select best members, decompose project tasks and distribute them among partners [12]. Other solutions are based on the use of the auction strategies in order to choose the best provider with the best criteria for a given service [13].

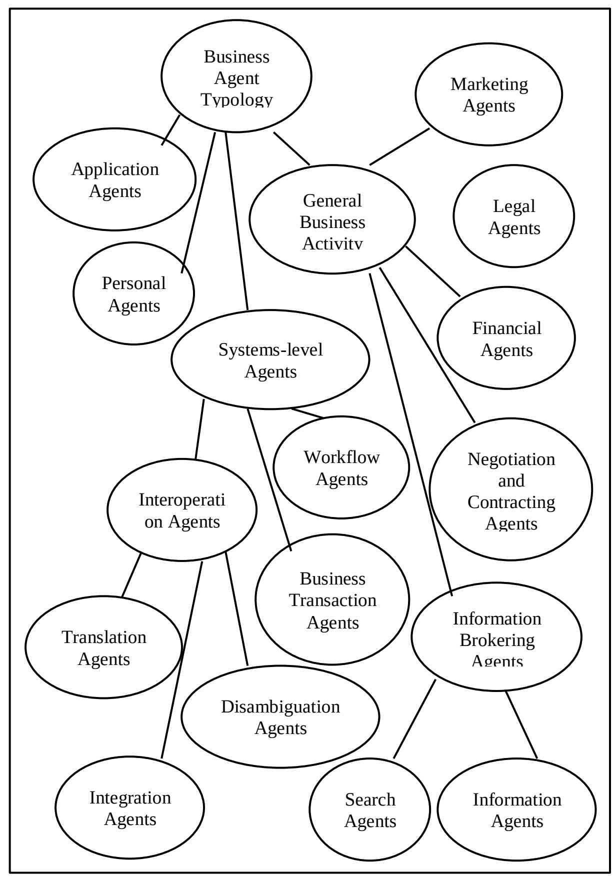

Figure 2. Agent typology in advanced IS and E-business domains[11]

Other research area of virtual enterprises development considers the idea of modelling the virtual enterprise as a set of distributed agents, where each agent represents an individual member. We can quote in this context AGORA architecture [14] where agents represent the partners in a VE and the interactions among the agents are supported by a specific Agent Interaction Protocol (AIP). Therefore, using agents as members in the virtual enterprises conduct us to focus on the negotiation process that was the concern of several works. Toshiya and Susumu consider a VE as agent with multi-utilities and contract-net protocol (CNP) applied as a negotiation algorithm amongst the agents [15]. In addition, Wang et al. offer a MAS solution where the VE initiator can be represented by either stationary or mobile agents to negotiate with VE partners. A hybrid multi-agent negotiation protocol is proposed to incorporate both the stationary and mobile agent negotiation phases to compose a more efficient and successful multilateral agent interaction regulation [16].

According to what was mentioned above, several proposals did not consider sufficiently the success of communication and interaction between the virtual enterprise partners. The problem is that each partner has his own services, expertise domain and data that are represented by their own selves. Yet, all these elements should be searched, invoked, transmitted and used by the virtual enterprise partners correctly in order to achieve the expected goals. Therefore, the use of ontologies can play a decisive role in this case because they bring very intrinsic contributions in the communication of complex systems. In this regard, Schneider and Hashemi believe that [17], ontology enhances decision and aid in knowledge management and discovery by helping people to better understand the complexity of big engineered systems, and by enabling integration among systems and data through semantic interoperability. Nowadays, ontologies are widely considered as a suitable solution for organizations to access, understand and manipulate existing data sources. Many organizations are heading towards the exploitation of the ontologies' expressiveness, the semantic interoperability and the ontologies inference' capacities to ensure their systems integration among their partners and to enhance their decision making processes [18].

Moreover, rich virtual enterprise interactions and services' exchanges between partners require a more loosely-coupled, peer-to-peer, dynamic and proactive mode of interaction. In this context, Web Services have the potential for ensuring this need by allowing the interoperability between companies among various software functioning on various platforms based on norms and open protocols. Recently, the "Everything as a Service" concept knew a remarkable success by supporting flexible integration of heterogeneous applications that led to the adoption of this concept in virtual enterprise construction [19].

Hence, our contribution is based on the premise that each partner offers his skills in the form of web services which will be orchestrated by agents using ontologies that facilitate the comprehension and ensure an efficient interaction.

In what follows, this work contribution that is based on the coupling between agents, ontologies and Web services is introduced.

3. THE PROPOSED MODEL

The proposed model is an agent-based architecture designed to create and manage virtual enterprises for specific goals (see Figure 3).

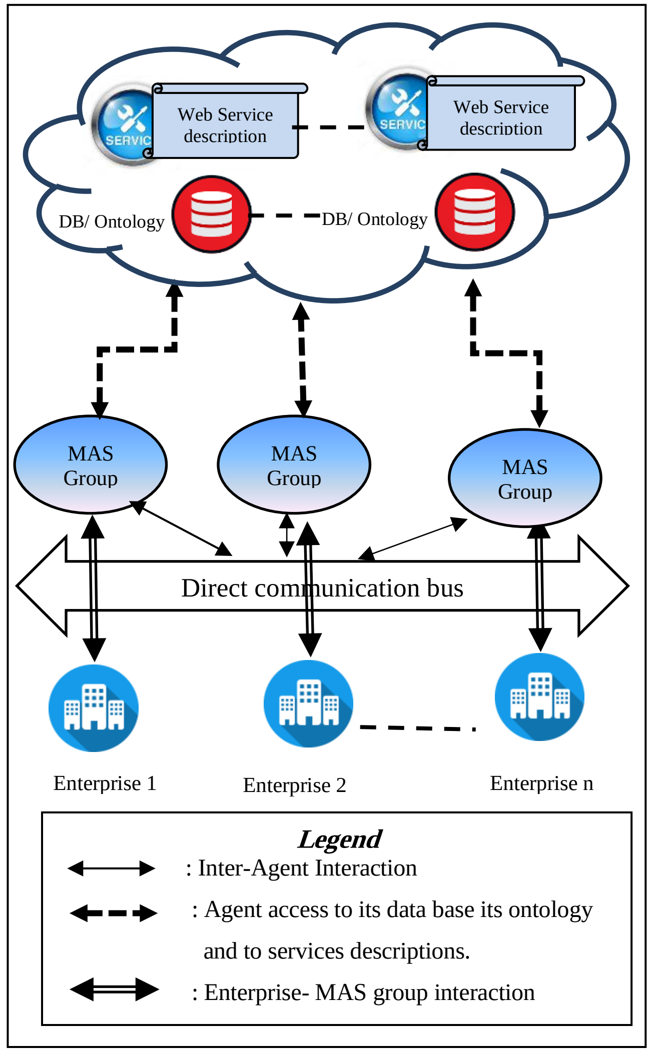

Figure 3. The Global Architecture

The model above is organized as a set of multi-agent system groups (MAS groups), which undergo the same structure and the same working mechanisms and provide some services. Each enterprise has its own MAS group which plays the role of an intermediary between it and the other enterprises by ensuring the necessary of interactions and by exploiting databases, ontologies and web services.

Because the enterprises must store and control a growing mass of information and services, the use of Cloud Computing technology is recommended. This latter's offers are multiplying to meet the growing needs; and organizations are increasingly ready to embrace this change and overcome their initial reluctance.

In the next section we will detail the components and operations of our model

A. Multi agent system group

A MAS group plays a vital role within each enterprise. It is an intermediary between its enterprise and the other enterprises by ensuring all inter-enterprise interactions in order to found a virtual enterprise for a given duration with clear and well-defined objectives (see figure 4).

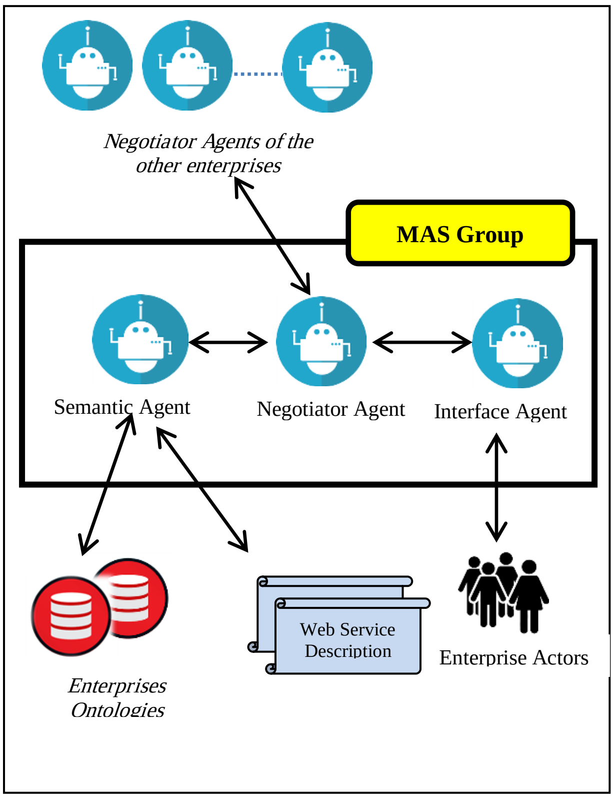

Figure 4. Multi Agent System Group

In order to achieve its objectives, the group of agents of each enterprise must consist of:

1) Interface Agent

This agent can be seen as a facilitator allowing the analysis of user data that are collected from a graphical interface. After the collection of information, a structured request will be created and sent to its negotiator agent. Generally, these data represent the service needs established by the enterprise's actors. The enterprise's actors of each company are, in fact, the first responsible for the determination of the web services offered by their enterprise for the external world and for the maintenance of their enterprise ontology.

2) Semantic Agent

The main role of the semantic agent is seeking the appropriate services required by its enterprise. After the formulation and launching of a new request by its company, the semantic agent compares the contents of this request with the ontologies of other enterprises. These last represent ontologies' domain where each one consists of a collection of terms, relations and definitions relevant to the business offered by its enterprise. These domain ontologies are expressed by the W3C Web Ontology Language (OWL) [20]. The choice of this language is motivated by the fact that it's a leader semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. Using these ontologies, the Semantic Agent can keep only the useful enterprises for a future collaboration. To complete its mission a natural language processing must be launched for comparing ontologies with the current request. At first, we parse domain ontologies for extracting the necessary terms for comparison. We use in this stage the Pronto API [21] which is a python module used to parse, create, browse and export ontologies in different formats including, of course, our used language OWL. After this stage, a tokenization job must be launched both on the outputs provided by Pronto API (extracted terms from the ontology) and on the request. At this level, the textual information is divided into individual words and some characters like punctuation marks or any excess white spaces are discarded. In addition, generic words of the language like determiners, articles and conjunctions are removed. After tokenization, the tokens of both the ontologies and the request are used for a similarity evaluation. This task is performed by using the Cosine Similarity Measure. According to our reading, this latter gives a useful measure of how similar two things are likely to be in terms of their subject matter [22]. In this case, the Semantic Agent must create two vectors: A and B with term frequency corresponding to the enterprise ontology and the current request. After the preparation of these two vectors, the similarity calculus is run according to the following equation:

$$\text{Similarity} = \text{Cos}(\theta) = \frac{\vec{A}.\vec{B}}{||\vec{A}||\ ||\vec{B}||}. \quad (1)$$

The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality (decorrelation), and in-between values indicating intermediate similarity or dissimilarity [23].

Therefore, to keep enterprises as possible future partners, the result of Cosine similarity must be strictly positive, which means that the initial request has an acceptable convergence degree with the ontologies of the kept enterprises and consequently has a strong possibility to be treated by these last.

From a technical point of view, to carry out the comparisons between ontologies and the current request after the Pronto's parsing, the Python environment is always used by the exploitation of the Python NLTK API

[24]. This latter is a leading platform for building programs to work with human language data. It provides a text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning that facilitate the Cosine similarity measure implementation.

After the determination of the future possible partners, the role of the negotiator agent starts by contacting them for choosing the right one after some negotiation tasks which will be clarified later in the negotiator agent section.

Another role provided by the semantic agent is to ensure the selection of the best web services proposed by their enterprises for a launched request. The selection task is soliciting in two cases:

- When the enterprise is requesting services: in this case, the proposed web services by the different bidders must be parsed by the semantic agent and compared with the initial request to select the suitable service and to choose the appropriate partner. Thereafter, the semantic agent informs its negotiator agent about the maintained partner.
- When the enterprise is providing services: based on the received request analysis, the semantic agent provides its negotiator agent with the most appropriate services for this request and the negotiator agent behaves as a bidder by offering these services to the requester.

The selection of appropriate web services is facilitated by their descriptions. At this level, we use OWL-S which is a DAML-based Web Services Ontology [25]. It is designed to support the Web Service related to the tasks of discovery, invocation, composition, interoperation and execution monitoring. The OWL-S is based on the use of three parts: service *profile* for advertising and discovering services; the *process model*, which gives a detailed description of a service's operation; and the *grounding*, which provides details on how to interoperate with a service, via messages.

The comparison process proceeds in the same way discussed before by using the different Python's API. By this process, we can extract all the necessary information's from an OWL-S file and compare them with a given request. Therefore, we keep or reject the service by application of these matching cases:

- Precise: the properties of the request correspond exactly to the service.
- Large: the service corresponds largely to the request. This case occurs when the request properties are included in the service properties. In this case, we solicit the enterprise actors to decide about this service.
- Partial: The service corresponds partially to the request. In this case, we solicit the enterprise actors to decide about this service
- Not adequate: The service is not adequate for the request. In this case, the service will be rejected.

3) *Negotiator Agent*

The negotiator agent constitutes the main component of its group. Its main role is facilitating communication, coordination and negotiation among its enterprise and the rest of partners. It is also responsible, when possible, for proposing the Web services provided by its enterprise to any requester by undergoing a bidding process. Also, the negotiator agent negotiates with other enterprises to obtain a service that is a satisfactory solution to the needs of its enterprise. To sum up, the negotiator agent plays both a supplier and a requester role and in order to accomplish its tasks, it employs its components that are clarified in the following figure (see Figure 5):

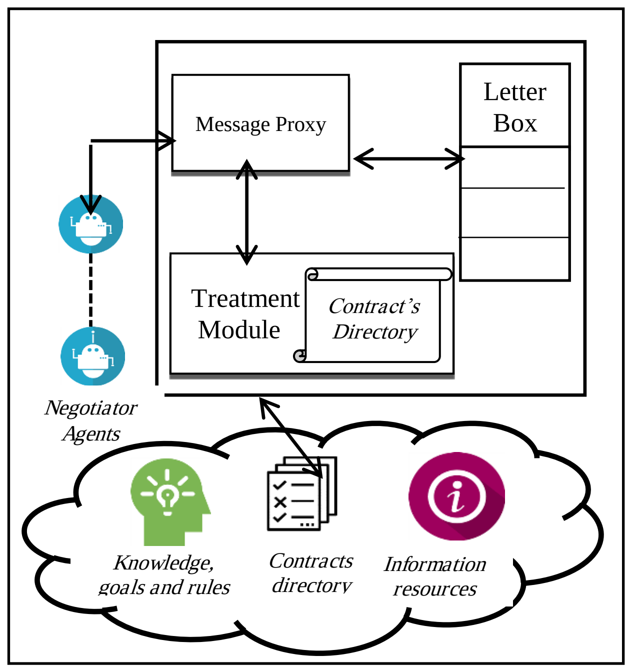

Figure 5. Negotiator Agent Structure

- Message proxy: It represents a channel or message support between negotiators. It contains all the protocols and mechanisms necessary to take and receive requests. All interactions between the agent and its environment pass through this module, which allows the negotiator agent to establish relationships with other negotiators in other systems. It communicates with other agents using FIPA ACL, and the FIPA messages. More details on this will be presented in the negotiation process section.

- Letter box: It's managed by the message proxy. When the messages arrive and while the agent is in a busy state, the communication module drops these messages into that mailbox. This box is a FIFO queue (First in First Out) used to store the messages in order to process them asynchronously.
- Treatment module: It ensures the management of the information necessary for the agent operation. The components of this module are the agent's knowledge base required to support the selection process, its goals on which a negotiator agent engages for a bidding or not and its rules base to support its decision making process and establishing a contract with the maximum benefits for its enterprise. The treatment module maintains also a contract's directory. This latter keeps the traceability of services previously carried out with the coordinates of their suppliers. This traceability can be used in the future to contact these suppliers directly for similar tasks without a biding call, unless the supplier refuses or proposes changes from the previous contract that does not suit the requester. Moreover, the treatment module contains additional information resources (variables, databases, files ... etc.) for ensuring the right operations of the negotiator agent.

B. Process of creation, operation and dissolution

The process of creation, operation and dissolution in our proposed model goes through the following phases:

1. After determining the needs and the objectives, the initial enterprise comes into contact with the other enterprises that can provide services to meet their needs. These enterprises can be discovered directly through the directory of contracts or by launching a call for biding for the first time.
2. Each bidder proposes its suitable service and the initial enterprise parse the OWL-S descriptions of these proposed services in order to keep the best and the more adequate service for its request.
3. The initial enterprise contacts the selected partners for accomplishing their contracts and to fix deadlines, prices and qualities.

Once the partners agree, a contract is established between the partners setting out the duties and rights of each partner.

5. The Virtual Enterprise starts to operate according to the specifications. As a result, the supervision and control tasks will be launched regularly with the possibility of suspending or inserting a new member into the virtual enterprise.
6. Once the entire work is completed, the virtual enterprise will be dissolved.
7. The initial enterprise keeps all the traces of the partners (contacts, skills, services, etc.) in order to exploit them in the future.

C. Negotiation process

The negotiation process is indispensable in the creation of the virtual enterprise with a given duration and clear and well-defined objectives. The core of our negotiation and interaction model is based on FIPA Contract Net Interaction Protocol [26]. Since this protocol doesn't include all the situations requested by our model, we add new situations and message exchanges expressed by the ACL language (Agent Communication language) [27] in order to obtain an enriched and complete protocol. Also, and for better communications, we use the message transport syntax for an XML based representation of ACL [27]. All messages expressed in ACL are sent in the XML body which gives to messages a form like this:

<msg-param>|receiver
| sender
| content
| language
| ontology
| protocol
Other Parameters
<msg-param>
<communicative-acts > |accept-proposal
| agree
| cancel
| confirm
| disconfirm
| inform
| propose
Other Performative
<communicative-acts>

For an interaction between two negotiator agents A and B of two different enterprises we distinguish two situations:

1. For a repeated need where a pair of negotiator agents A and B have already worked together and the negotiator agent A has the sufficient information about the provider agent B (information about the service and its provider agents is deposited in the Negotiator's Contracts directory). In this case, Agent A before it makes its request, it consults its directory of contracts and focuses negotiator agent B which satisfies its needs in order to make this request. The negotiator agent B can respond with three options:

- It agrees to treat the request as it is and with the maintaining of the old conditions saved in the contract directory.
- It refuses to process the request for one reason or another
- It agrees to treat the request but with new conditions. In this case agent B informs agent A about these changes, and this last decides to accept these new conditions or to launch a new call for proposal.

With the UML sequence diagram and the FIPA ACL language we can present this scenario as follows (see Figure 6).

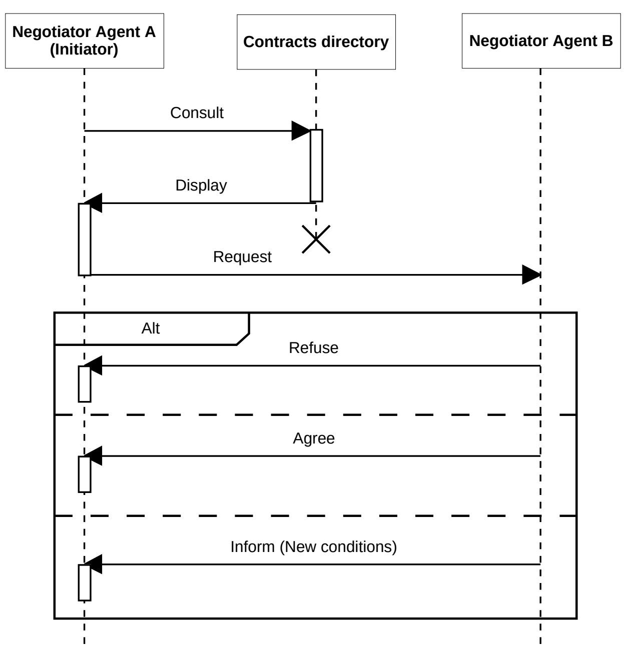

Figure 6. Sequence Diagram (Situation 1)

2. In case of failure in the precedent scenario or for a new need that was not previously established, we make a new call for bidding (Call for Proposal) and we find ourselves in the situation where negotiator agent A (requester) and the other negotiator agents (probable partners) interact with each other for the first time to satisfy this need. In this case the bidder negotiator agents propose to the negotiator agent A their best found services satisfying the requested criteria with their conditions (price, deadlines, quality ….).So, the Agent Negotiator A and through its semantic agent selects the best bidder as a future partner. Afterwards, it sends to this selected partner a confirmation of acceptance of its offer (Accept Proposal) and a rejection notification for the other bidders (Refuse Proposal). If this phase goes well, negotiator A records information and properties about the request and the selected partner in its contracts directory to use them directly in the future in similar cases. This scenario can be presented in UML sequence diagram as follows (see Figure 7):

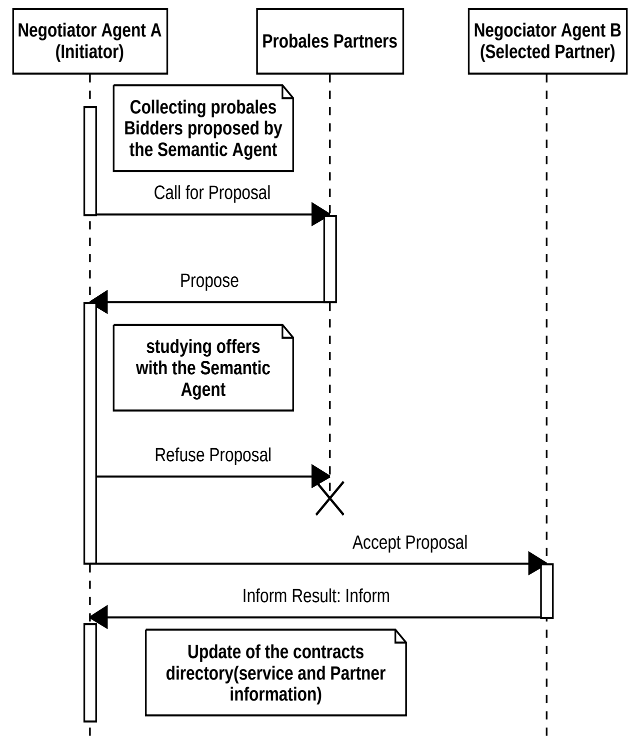

Figure 7. Sequence Diagram (Situation 2)

4. CONCLUSION

In this paper, an agent's model for establishing virtual enterprises was introduced. The core of this model is based on the use of ontologies and web services. We consider in this model all the steps related to the virtual enterprises lifecycle.

The proposed model is essentially organized as a set of agent groups which undergo the same working mechanisms and which represent the different partners of a virtual enterprise. The main tasks in this model are the discovery of partners and the establishing of a negotiation process between them. The selection of partners is ensured by the exploitation of ontologies and Web services description. The negotiation process is performed by an improved protocol through the use of FIPA Contract Net Interaction Protocol and ACL language. We have given in this paper some technical tips for the implementation of this model by exploiting the Python environment. Finally, we report that most of the developers of the multi-agent systems are familiarized with Java environments, so, there are several solutions to integrate their solutions in Python environment and therefore in our model and this by using some APIs like JPype [28], Jython [29] or Py4J [30].

REFERENCES

[1] M. M. Cruz-Cunha, and G. D. Putnik, " Market of Resources: A Cost and Effort Model". In: G. D. Putnik and M. M. Cruz-Cunha, ed., *Encyclopedia of Networked and Virtual Organizations*, 3rd ed. IGI Global, pp.891-898, March 2008

[2] L. M. Camarinha-Matos, and H. Afsarmanesh, "Virtual Enterprise Modeling and Support Infrastructures: Applying Multi-agent System Approaches", M. Luck et al. (Eds.): ACAI 2001, LNAI 2086, pp. 335-364. 2001

[3] L.S.Bahram, N.Shahrzad , M.O.Ahmet , Ö.Ü. Hakki, D.Erdogan , E.K. Sadik, and A. Wassenaar, "An ontology-based multi-agent virtual enterprise system (OMAVE): part 2: partner selection" International Journal of Computer Integrated Manufacturing 30(10):1072-1092 · January 2017.

[4] H. Kang, F. Dai, and Z. Liang, "A web service-based virtual enterprise model"Proceedings of the International Conference on Electrical and Control Engineering. Yichang, China. 2011.

[5] A.Samdantsoodol, "An Adaptive Framework for Improving the Effectiveness of Virtual Enterprises in the Supply Chain", A thesis submitted in partial fulfilment of the requirement of Staffordshire University for the degree of Doctor of Philosophy, pp. 28-29. February 2017

[6] PMI. "A Guide to the Project Management Body of Knowledge". 5th ed. Pennsylvania: Project Management Institute, Inc, 2013.

[7] L. Ferreiraa, N. Lopesa, P. S. Ávilab, H. Castroa, M. L. R. Varelac,d, G. D. Putnikc,d, R. Martinhoe,f, R. Rijof, I. M. Mirandag, M. and M. Cruz-Cunhaad, " Virtual Enterprise integration management based on a Meta-enterprise – a PMBoK approach ", Procedia Computer Science 121 (2017), pp. 1112–1118, 2017

[8] N. Mouhib, S. Bah, and A. Berrado, "Enterprise architecture improvement for virtual enterprise modeling", Proceedings of the International Conference on Industrial Engineering and Operations Management Rabat, Morocco, April 11-13, pp. 2216 - 2227, 2017

[9] L.M. Camarinha-Matos, and H. Afsarmanesh, "The Virtual Enterprise Concept" in Infrastructures for Virtual Enterrises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, pp. 3-30, 1999

[10] M. Wooldridge, " An introduction to multi-agent systems". New York: John Wiley & Sons.2002

[11] M.P. Papazoglou,"oriented technology in support of e-business". Communication of the ACM, 44(4). 2001

[12] Z. Yang, H. Lin and C. Liu, "Analysis of the Virtual Enterprise Partner Selection based on Multi-Agent System," Journal of Communication and Computer, vol. 6, no. 8(Serial No.57), pp. 49-53, Aug 2009.

[13] B. Lotfi Sadigha, F. Arikanb, A. M. Ozbayoglub , H. O. Unverc, and S. E. Kilicd," A Multi-Agent System Model for Partner Selection Process in Virtual Enterprise", Procedia Computer Science 36 (2014) 367 – 372, 2014

[14] S.A. Petersen, J. Rao1, and M. Matskin," Virtual Enterprise Formation with Agents – an Approach to Implementation", IEEE/WIC International Conference on Intelligent Agent Technology, IAT, pp 527 – 530, 2003

[15] K. Toshiya, and F. Susumu, "A Proposal on Negotiation Methodology in Virtual Enterprise", Collaborative Business Ecosystems and Virtual Enterprises, pp 125-132, 2002

[16] G. Wang, T.N. Wong, and X. Wang," A hybrid multi-agent negotiation protocol supporting agent mobility in virtual enterprises", Information Sciences—Informatics and Computer Science, Intelligent Systems, Applications: An International Journal, Volume 282, pp 1-14, October, 2014

[17] Schneider, T.Hashemi, A.Bennett, M.Brady, M.Casanave, C. Graves, and H. Yim, "Ontology for Big Systems: The Ontology Summit 2012 Communiqué". Applied Ontology. 7(3): 357-371. 2012.

[18] A. ABADI, A. H. BEN-AZZA, S. SEKKAT, and E. ZEMMOURI, "An ontology-based framework for virtual enterprise integration and interoperability". Proceedings of the 2nd International Conference on Electrical and Information Technologies. ICEIT'20 16. Tangier, Morocco. 2016.

[19] G. Li, and M. Wei, "Everything-as-a-service platform for on-demand virtual enterprises". Information Systems Frontiers, Volume 16(3), pp 435-452, 2014

[20] *Web Ontology Language (OWL)*. [online] Available at: https://www.w3.org/OWL/.

[21] GitHub. (2018). *althonos/pronto*. [online] Available at: https://github.com/althonos/pronto.

[22] Singhal. A, "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 24 (4): 35–43. 2001

[23] N. D. Sonal, and R. D. Ratnadeep," Cosine Similarity for Substituted Text Detection". International Journal of Advanced Research in Computer Science and Software Engineering. 4(1):903 -907. 2014

[24] Nltk.org. (2018). *Natural Language Toolkit — NLTK 3.4 documentation*. [online] Available at: https://www.nltk.org/.

[25] W3.org. (2018). *OWL-S: Semantic Markup for Web Services*. [online] Available at: https://www.w3.org/Submission/OWL-S/.

[26] Fipa.org. (2018). *Fipa contract net interaction protocol specification*. [online] Available at: http://www.fipa.org/specs/fipa00029/.

[27] Fipa.org. (2018). *FIPA ACL Message Structure Specification*. [online] Available at: http://www.fipa.org/specs/fipa00061/SC00061G.html.

[28] Jpype.sourceforge.net. (2018). *JPype - Java to Python integration*. [online] Available at: http://jpype.sourceforge.net/.

[29] Jython.org. (2018). [online] Available at: http://www.jython.org/downloads.html .

[30] Py4j.org. (2018). *Welcome to Py4J — Py4J*. [online] Available at: https://www.py4j.org.

Dr. Mahmoud BRAHIMI received his engineering degree in 1998, his magister degree in 2001, and his PHD degree in Computer Science in 2009 from the University of Constantine in Algeria. Now, he is a lecturer at computer science department at the University of M'sila in Algeria. His main areas of research interest are Advanced Information Systems, E-Business and Multi-Agent Systems. He is a reviewer of a number of journals and conferences.