Automatic Domain Ontology Construction Mechanism

Aarti Singh
Associate Professor, MMICT&BM
Maharishi Markendeshwar University, Mullana
Haryana, India
singh2208@gmail.com

Poonam Anand
Research Scholar, MMICT&BM
Maharishi Markendeshwar University, Mullana
Haryana, India
poonambhatia02@gmail.com

***Abstract*— Ontologies play vital role in agent communication in present semantic web. Manual construction of ontology is very complex and time consuming process. Thus it is mandatory to automate this task. A number of tools have been created to help the ontology generation in a semi automatic or manual way. But there is no standard framework for complete automation of this process. Though ontologies are highly reusable so based on a basic ontology other ontologies may be built for specific application domain. This work attempts to present a mechanism for automatic construction of ontology using intelligent agents.**

Keywords— Ontologies; Ontology construction; Agent communication; Semantic web

I. INTRODUCTION

Ontology refers to the shared understanding of a domain of interest, which is often conceived as a set of classes (concepts), relations, functions, axioms and instances (Gruber,1993). Ontology is vocabulary describing the terms in a domain with motive of providing clear understanding of domain under consideration; it is specifically meant for knowledge transfer to machines [5]. Ontologies have their roots in artificial intelligence, and their emergence was led by human desire to make machines intelligent. In order to make software component intelligent, it must be supported with knowledge about its environment or domain. It was accomplished through Ontology which is vocabulary of a domain, providing clear and concise description of various concepts, terms, objects, attributes and their inter-relationships. It is only because of ontologies that promising, cooperative and autonomous software agents could be implemented and deployed in web based applications. Intelligent software agents are small software entities essentially supported by an ontology, which makes them capable of understanding their environment and work autonomously without human assistance. Presently agent technology is widely been employed in many applications of autonomous nature, such as sensor networks, information extraction from semantic web etc.[13] With increased scope of artificial intelligent components, need of supporting ontologies is also increasing.

Developing ontology is a time consuming and laborious task. There are two possible ways for developing ontology:

- From scratch
- From existing ontologies

Constructing ontology from scratch requires great time and efforts. The later option of developing ontologies from already existing ontologies provides an appealing alternative. Reusable domain ontologies provide opportunities for researchers to exploit and reuse existing domain knowledge to build their applications with much ease and reliability, along with cost effectiveness. However, discovering the already existing, relevant ontologies is also a challenging task for the users. Considering the fact that ontologies are mainly employed by agents in their communications, they may be the key for finding relevant ontologies from semantic cyber space. This provided us the motivation for presenting framework for automatic ontology generation. The major contribution of this work is that present framework would amalgamate relevant parts of ontologies existing in homogeneous or heterogeneous domains while constructing desired ontology.

Rest of the paper is organized as follows; section 2 presents the existing literature in the field. Section 3 elaborates the proposed work. Section 4 concludes the paper.

II. RELATED WORK

Ontology building is very crucial process in the field of ontology development. This process requires detailed analysis of concepts, terms and relationships in domain of interest. It demands minute transfer of knowledge associated with concepts since later it is deployed for clear understanding of the domain. Manual construction of ontology from scratch is very complex and tedious job, so it is necessary to automate this process. This section presents the work of eminent researchers in this direction.

Blomqvist E. [3] proposed method for automatic construction of ontology by using ontology design pattern and pattern matching mechanism. Dahab M.Y. et al. [7] discussed the tool 'TextOntoEx' which constructs the ontology from natural domain text using semantic pattern based approach. But it is only limited to instances of known relations and doesn't discover new relations. Moreover constructing semantic pattern is also time consuming process. Van A.C. et al. [20] presented an approach for creating and designing ontology for agent communication using Protégé 2000 and JADE framework. Fortuna B. et al. [10] proposed OntoGen tool for semi automatic construction of ontologies by using machine learning and data mining algorithms. Hendler J. [13] integrated the ontologies and agent technology and discussed the mapping between agents and ontologies.

Sabou M. [17] described a semiautomatic method to extract domain ontologies from software documentation using Application Programming Interface (API) tools. Boyce S. et al. [4] developed the content ontologies for education domain. Wache H. et al. [21] reviewed the use of ontologies for integration of heterogeneous resources. The languages used for representing ontologies and ontology mapping techniques are also being evaluated and compared. Chandrasekaran B. et al. [5] described the conceptual introduction to ontologies and their role in information system and Artificial Intelligence (AI). The clarification of domain structure of knowledge and knowledge sharing with help of ontologies are also discussed.

Noy N. F. et al. in [15] discussed the ontology development methodology for declarative frame based systems. Kalaivani P. et al. [14] described framework for construction of poultry domain ontology using protégé. Haav H.M. [12] discussed the framework for automatic construction of domain specific ontologies using Natural Language Processing (NLP) and Formal Concept Analysis (FCA). Singh A. et al. [18] discussed the Contract Net Protocol (CNP) and introduced Trust Establishment Protocol (TEP) in CNP. Cimiano P. et al. [6] discussed the state of art in ontology learning with respect to ontology learning subtasks. They also discussed the applications, challenges, and trends in field of ontology learning research.

Pinto H. S. [16] presented the methodologies used for ontology creation from scratch and techniques, guidelines and methods that are helpful in construction of ontology. The challenges in ontology building also discussed in this paper. Ding Y. et al. in [8] described ontology generation and various methods for generation of ontologies in automatic or semi-automatic manner. The methods of concept extraction, ontology reuse, relation extraction are also discussed. Bedini I. et al. [2] presented the framework for evaluating the automation of ontology generation and compared the existing software geared towards automatic ontology generation but failed to develop ontology with Extensible Mark-up Language (XML) files using available softwares.

Singh A. et al. [19] proposed an intelligent interface and mapping framework that provides mapping for homogenous and heterogeneous domain ontologies. Wooldridge M. et al. in [22] reviewed the issues associated with theory and applications of intelligent agents. Fernández-López M. et al. [9] described and analyzed the various methodologies for building ontologies. Alani H. [1] presented approach for construction of ontology automatically by reusing the online ontologies but this is restricted only to small size or medium size of ontologies.Gruber T. R. [11] described the mechanism for defining ontologies that are portable over representation system.

From the above literature review it is clear that need for automatic ontology construction mechanism had been felt by research community. Many researchers have put in their efforts in this direction however there is no such standard mechanism existing for this purpose as of now. This provided us the motivation for the present work. Next section elaborates the proposed framework.

III. PROPOSED WORK

The proposed framework mainly comprises of User Interface Agent (UIA), Ontology Mapping Agent (OMA), Ontology Generation Agent (OGA) and Manager Agent (MA). Every agent is responsible for a specific task and overall goal is to automate domain ontology generation for the user. This mechanism is henceforth termed as Automatic Domain Ontology Construction Mechanism (ADOCM). This mechanism employs intelligent agents because they possess autonomous behavior and can carry on designated job without human intervention. The high level view of this mechanism is shown in fig. 1. Description of various agents is as follows:

- ***User Interface Agent (UIA):*** This agent serves as the interface between ADOCM and the user. It interacts with users and accepts the domain and context in which ontology is to be constructed. It is supported with an ontological agent list, which maintains history of agent-id and domain of their ontology. The purpose of this list is to have faster access of relevant ontologies through agents. It also interacts with OGA and MA and provides their responses to the user appropriately.
- ***Manager Agent (MA):*** It interacts with UIA and receives the domain of required ontology. UIA also sends agent_ids to MA if found in OHL. MA advertises the requirement of relevant ontologies to all agents in the cyber space (termed as Ontology agents (OA) here). Once the agent working with relevant ontology respond to MA, their ontology schema may be requested. MA also communicates with Ontology Generation Agent (OGA) by passing ontology schema to it.

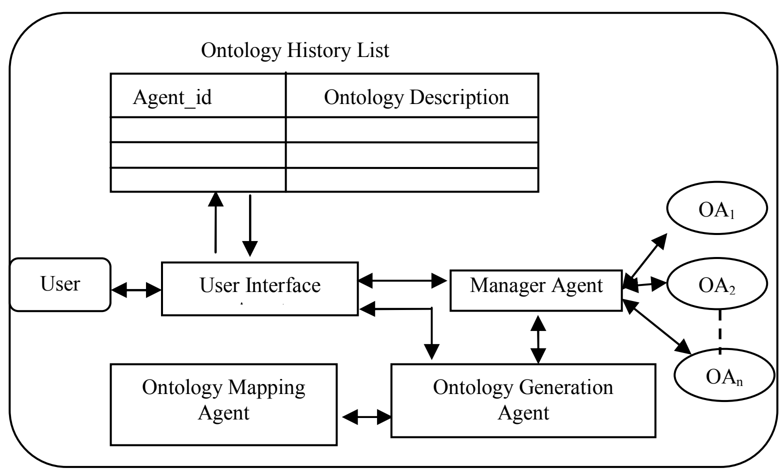

Fig. 1 High level View of ADOCM

- ***Ontology Generation Agent (OGA):*** This agent is responsible for generating the actual ontology according to user's requirements. UIA provides the concepts required by the users to OGA and relevant ontology schemas are provided by the MA. It also interacts with the ontology Mapping Agent for mapping between various ontologies.

- ***Ontology Mapping Agent (OMA):*** This agent facilitates mapping among ontologies belonging to homogeneous or heterogeneous domains. While generating ontology automatically, the volunteer agents may be from different domains of applications, even if they belong to same domain, they might have been developed by different persons with different terminology for same things. Ontology mapping becomes essential while dealing with various agents of different origin. It is also vital in our framework and OMA provides the functionality to map different terms having similar meaning and also help in deciding subset and superset of concepts employing intension or extension of ontology. Intension and extension are described as follows [19]:
  - **Intension:** It can be defined as the subsumption of sets of instances classified under the concept reflecting the $\subset_c$ relation.
  - **Extension:** Extension of a concept is the set of instances that can be classified under that concept. The extension of the universal concept (everything) Cu is the set of all instances while the extension of the absurd concept (nothing) $\perp$ C is the empty set.

Next subsection provides the detailed working of ADOCM.

A. Working of ADOCM

The various agents described above work in close coordination with each other. Initially a user interacts with UIA and provides the domain and context for ontology construction. On receiving the domain name, UIA searches the OHL to find agents using similar ontologies. If one or more match is found UIA sends relevant agent_id to MA. In case, if the OHL is empty or does not contain any agent using ontology of interest, then UIA sends the domain name to MA. MA then advertises the requirement for the ontology in the cyber space using FIPA standardized CNP [12]. On receiving response from some agents using relevant ontologies, MA requests for concept list of their ontologies. MA provides this concept list to UIA. UIA in turn enters this data in OHL for future usage and also presents this concept list to the user for selection of appropriate concepts required in target ontology. The terms and concepts approved by the user are passed to the OGA for further processing. Once the concepts are being approved by the user, MA fetches ontology schemas from the respective agents through OA. The ontology schemas that are fetched by various ontology agents are also provided to OGA. However ontologies being fetched may be from homogenous or heterogeneous domains, since here our concern is to construct target ontology by using relevant parts of existing ontologies. Also, different existing ontologies might be developed using different tools/languages and might be focusing on different aspect of a domain, thus ontology mapping mechanism is required while generating target ontology. OMA agent in proposed framework is deploying ontology mapping mechanism proposed in [19] for establishing compatibility of reuse process. OMA maps the concepts from various ontologies and provides target ontology to OGA which then passes it to UIA for verification by the user.

Next subsection provides the wok flow of ADOCM.

B. Flow Diagram of ADOCM

The flow diagram of ADOCM is shown in the figure 2. This diagram illustrates the sequence of steps being followed while automatic generation of domain ontology using intelligent agents.

1. User requests for ontology generation by providing the domain name to UIA.
2. UIA searches for relevant ontology concepts and associated agent_id from OHL. This list contains agent_id as well as description of their ontologies. If a match is found in OHL then corresponding agent_id is fetched and passed to MA for further processing.
3. In case if OHL is empty or doesn't contain relevant ontologies, the requested domain name is passed to MA by UIA. However if some matching agent_ids are retrieved from OHL they are passed to MA.
4. If MA receives agent_ids from UIA, it directly requests those agents for concept list of their ontologies. Otherwise MA advertises ontology requirement for the requested domain and waits for responses from the agents existing in cyberspace. On receiving response from some agents it requests for concept list of their ontologies. Concept list to ontology agents. This is done with the help of CNP. . OS along with their respective agent_id is provided to the MA. MA provides concept list and agent_id to UIA.
5. UIA inserts this information in OHL for future usage.
6. UIA then provides the concept list to user for selection of appropriate concepts in target ontology
7. User filters the list for appropriate concepts as per requirement and provides approved concept list to UIA.
8. UIA passes this approved concept list to OGA for generation of target ontology. For this purpose OGA requires actual coding of existing ontologies. To

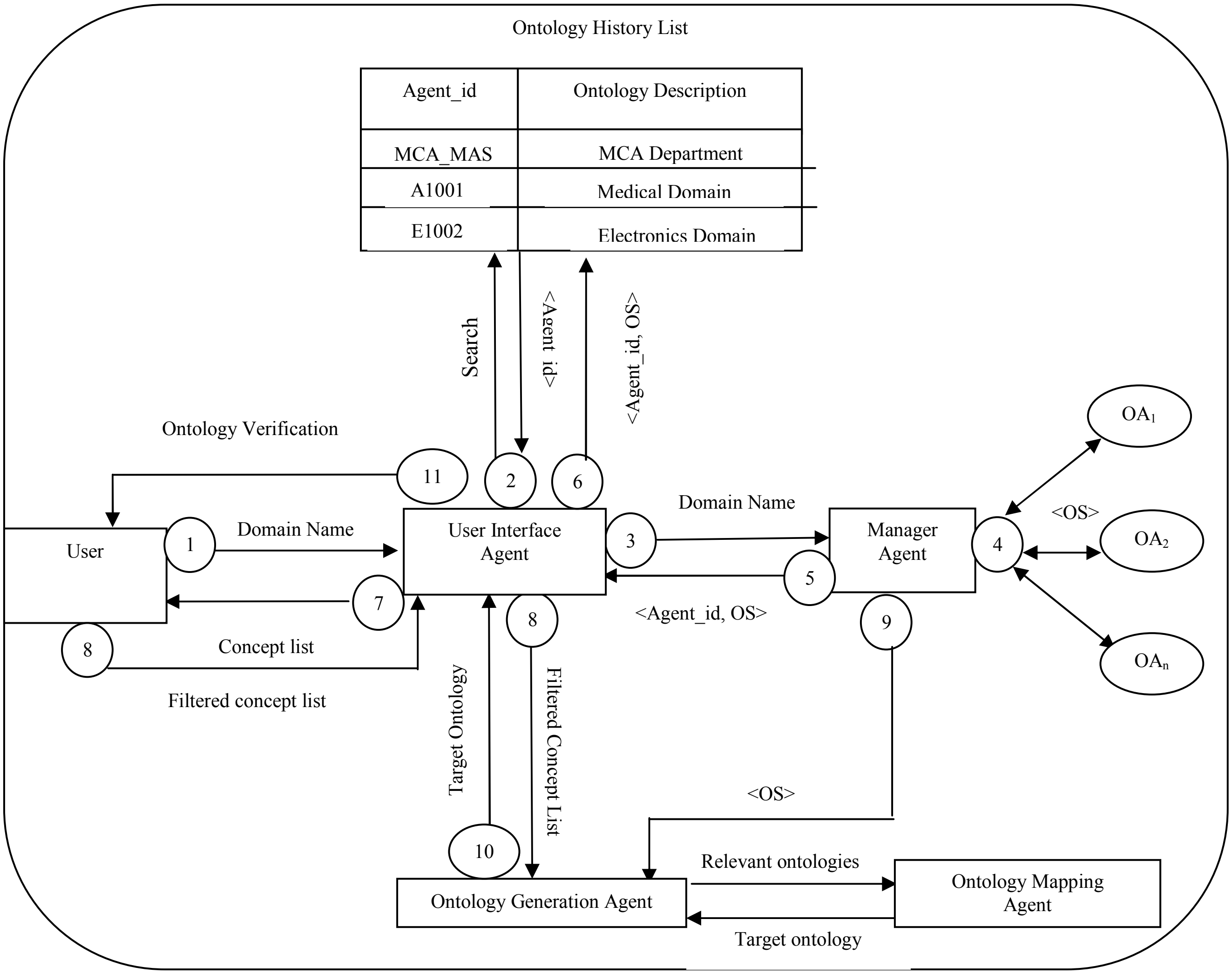

Fig.2 Flow Diagram of ADOCM

accomplish this MA fetches Ontology schema from the associated agents and passes it to OGA.

9. Now OGA and OMA work in close coordination for building target ontology. OGA starts creating the hierarchical structure in the form of base_class and its sub_classes. However this relationship is being derived using extension and intension properties of the concepts, being applied by OMA. (for details of Ontology mapping mechanism being adopted by OMA see [19]). Ontology mapping is essential here since concepts are being collected from diverse sources focusing on different aspects of different applications, it's necessary to ensure their mapping sequence.

10. OGA provides the target ontology to UIA.

11. Finally, UIA forwards target ontology to user for verification of concepts and its validation.

The algorithms for various agents are discussed in next subsection.

C. Algorithms

Algorithms for various agents used in ADOCM are discussed as follows:

```
Input: Domain_name for ontology construction
Output: concept_list, generated ontology
Actions: accept, search, update
{
accept (domain_name) ←—— User
if (OHL!=NULL)
{search (domain_name, OHL) }
if(match found)
{return agent_ids to MA;}
else{
call (MA (domain_name))
receive <agent_id, OS>
update (OHL) }
return <concept_list> to user and store the approved concepts
in FCL;
send <FCL> ——→ OGA
receive <generated ontology> ←——OGA
verify <generated ontology> from user;}
```

Fig. 3(a) User Interface Agent

```
Input: Domain_name , agent_id
Output: OS, agent_id
Actions: receive(domain_name), generate(CFP)
{if( inut==domain_name)
  {receive (domain_name) ← UIA;
 generate (CFP) for domain_name (ontology);
 receive (OS) ← OA's
 return <agent_id, OS> → UIA
  }
else if(input==agent_ids)
{ send request for ontology_schema(agent_ids)
  return <agent_id, OS> → UIA}
  else if request for ontology_schema(agent_id)
    return <ontology_ (agent_id)> → OGA;
}
```

Fig. 3(b) Manager Agent

```
Input: FCL, Agent_id
Output: Target ontology
Actions: call, receive
{
receive (FCL) ← UIA;
receive (ontology_schema) ← MA
call(OMA(ontology_schema(agent_id1,agent_id2))
receive (concept mapping from OMA)
create target_ontology
return <target_ontology> → UIA ;}
```

Fig. 3(c) Ontology Generation Agent

```
Input: <ontology1, ontology2>
Output: concept mapping
Actions: accept, return
{
accept <ontology1, ontology2> ← OGA;
pick(c1(o1) & c2(o2))
if (intension(c1,c2)==true)
        return (c1 is subset of c2)
else if (extension(c1,c2)==true)
        return (c1 is superset of c2)
  else return (c1==c2) ;}
```

Fig. 3(d) Ontology Mapping Agent

D. Case Study

Suppose that the user want to build the ontology for "electronics_equipments" domain. User provides this domain name to UIA. The concept list of the various ontologies of same domain is searched from OHL. If not found the following concept list is generated by the respective OA's. The concept list shown in Table 1 is provided to user for filtration of concepts. User provide filtered concept list to UIA which is shown in Table 2.

Table 1: Initial Concept List

Photo_camera	Personal_computer	Mobile_phone
Microprocessors	Laptop	Digital_camera
Accessory	Camera_and_photo	PC_Board
Accessories		

Table 2: Filtered Concept List

Photo_camera	Personal_computer	Digital_camera
Microprocessors	PC_Board	Mobile_phone
Accessories		

Now this filtered concept list is given to OGA for generation of ontology. OMA uses intension and extension mechanism on these concepts as:

(i) Intension

Personal_computer, Photo_camera, Mobile_phone$\subset$ Electronics_equipments.

(ii) Extension

'Digital_camera' extends the features of 'Photo_camera'. Extension is required when we need to communicate in reverse direction.

So after applying mapping mechanism following ontology is produced:

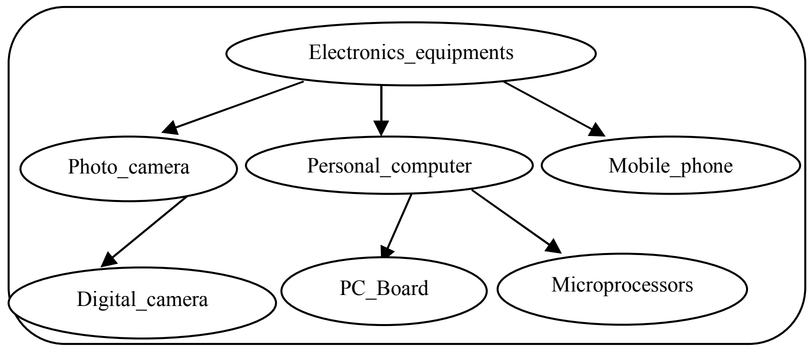

Fig. 4 Domain Ontology for Electronics_equipments

This target ontology is passed to UIA and then finally to user. This process is repeated as per requirements of user for building of domain ontology.

IV. CONCLUSION

In this paper, mechanism for automatic generation of domain ontology using intelligent agents is presented. This mechanism can automate the ontology generation process and would significantly reduce the time and effort required in this process. This mechanism completely automate the ontology building process, the major contribution is facility to map concepts from already existing homogenous or heterogeneous domain ontologies thereby facilitating reuse of relevant parts of any existing ontology. The implementation and evaluation of this mechanism is still under progress.

REFERENCES

[1] Alani H.: Position paper: ontology construction from online ontologies. In Proceedings of the 15th international conference on World Wide Web. ACM.pp.491-495, May 2006.

[2] Bedini I. & Nguyen B.: Automatic ontology generation: State of the art. PRiSM Laboratory Technical Report. University of Versailles, 2007.

[3] Blomqvist, E.: Fully automatic construction of enterprise ontologies using design patterns: Initial method and first experiences. In On the Move to Meaningful Internet Systems

2005: CoopIS, DOA, and ODBASE, pp. 1314-1329. Springer Berlin Heidelberg, 2005.

[4] Boyce S. & Pahl C.: Developing domain ontologies for course content. Educational Technology & Society-ETS, vol. 10(3), pp.275-288, 2007.

[5] Chandrasekaran B., Josephson, J. R., & Benjamins, V. R.: What are ontologies, and why do we need them?. Intelligent Systems and Their Applications, IEEE, vol.14 (1), pp.20-26, 1999.

[6] Cimiano P., Völker, J., & Studer, R.: Ontologies on demand? a description of the state-of-the-art, applications, challenges and trends for ontology learning from text, 2006.

[7] Dahab, M. Y., Hassan, H. A., & Rafea, A.: TextOntoEx: Automatic ontology construction from natural English text. Expert Systems with Applications, vol. 34(2), pp. 1474-1480, 2008.

[8] Ding, Y., & Foo, S.: Ontology research and development. Part 1-a review of ontology generation. Journal of information science, vol. 28 No.2, pp.123-136, 2002.

[9] Fernández-López M., & Gómez-Pérez A.: Overview and analysis of methodologies for building ontologies. The Knowledge Engineering Review, vol. 17(2), pp.129-156, 2002.

[10] Fortuna, B., Grobelnik, M., & Mladenić, D.:Semi-automatic data-driven ontology construction system, 2006.

[11] Gruber T. R.: A translation approach to portable ontology specifications. Knowledge acquisition, vol. 5(2), pp.199-220, 1993.

[12] Haav, H. M.: An application of inductive concept analysis to construction of domain-specific ontologies. In Emerging Database Research in East Europe, Proceedings of the Pre-Conference Workshop of VLDB, pp. 63-67, Nov. 2003.

[13] Hendler J.: Agents and the semantic web. Intelligent Systems, IEEE, vol.16 (2), pp.30-37,2001.

[14] Kalaivani, P., Anandaraj, A., & Raja, K.: An Ontology Construction Approach for the Domain of Poultry Science Using Protégé. ArXiv preprint arXiv: 1302.5417,2013.

[15] Noy N. F., & McGuinness, D. L.: Ontology development 101: A guide to creating your first ontology, 2001.

[16] Pinto H. S. & Martins J. P.: Ontologies: how can they be built?. Knowledge and Information Systems, vol. 6(4), pp.441-464, 2004.

[17] Sabou, M.: Extracting ontologies from software documentation: a semi-automatic method and its evaluation, 2004.

[18] Singh, A., Juneja D., & Sharma A. K.: Introducing Trust Establishment Protocol in Contract Net Protocol. In Advances in Computer Engineering (ACE), 2010 International Conference IEEE, pp.59-63, 2010.

[19] Singh, A., Juneja, D., & Sharma, A. K.: Design of an Intelligent and Adaptive Mapping Mechanism for Multiagent Interface. In High Performance Architecture and Grid Computing pp. 373-384. Springer Berlin Heidelberg., 2011.

[20] Van Aart, C., Pels, R., Caire, G., & Bergenti, F.: Creating and using ontologies in agent communication. In Proceedings of the Workshop on Ontologies in Agent Systems, July 2002.

[21] Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S.: Ontology-based integration of information-a survey of existing approaches. In IJCAI-01 workshop: ontologies and information sharing, vol. 2001, pp. 108-117, 2001.

[22] Wooldridge M.& Jennings, N. R.: Intelligent agents: Theory and practice. Knowledge engineering review, vol. 10(2), pp.115-152, 1995.