A Framework of SDML Based Reusable Web Page for Semantic Web

Herald Noronha[1]* Dr. Vijay Prakash Agrawal[2]

[1]Research Scholar, CMJ University, Shillong

[2]Associate Professor, Department of Math, BSA, PG College, Mathura

Abstract – Web based development demands not only quick and reliable web page designing but also designing must be reusable. Generally, lot of developer time and efforts are used in developing solutions for difficult, time consuming and complex web based systems. One of the solutions is the web based designs document. Design process takes maximum time and efforts and is never reused in the future. Reusability is achieved up to some level in coding practices like OOPs, Component based developments, Active-X, technology, where a piece of written code is reused after passing through few checks for non-discloser of the blueprints of the solution.

In present paper an attempt has been made to addresses the methodology for achieving the web page design reusability of a qualitative web based system and effort minimization by applying the inference on the stored design documents.

Key words: Semantic Web, HTML, SDML

----------♦----------

INTRODUCTION

Design phase is considered to be the most important phase to achieve reusability in web based projects, as it bridges the gap between requirement phase and the development phase. This includes functional and user interaction design. If a web based design is made reusable, the reusability in the development phases can be achieved because the coding phase is very much driven from the software design. Reusable web based design methodology is of special concern with the effort minimization, which could be achieved by following this approach. This paper also discusses the notation of the design document storage for reusability. This approach will provide good domain solution in minimum efforts. A good review mechanism can also be imposed on the stored reusable web based design for assigning quality attributes.

Proposed approach eliminates the need for fresh efforts for web based design every time a solution is required. The web based design document having diagrams, images will be stored in textual format and every design stored will have few keyword and properties. Keyword submission per design facilitates the search against the specified keyword and outcome of the result having the attribute tag of quality benchmark, like number of times the design is reused. The research also includes one tool, which helps in maintaining the central repository of the different design documents and inference mechanism on the stored design repository.

SDML Based Web Page Design Representation for Semantic Web

The reusability of web page design for a software system could be achieved by storing the graphical design documents in the form of text. The graphics-symbols of the design document can be represented in the form of text. Tools are available which allow the user to draw the web page with diagrams. These tools however lack one aspect to store the design elements in the form of structured text as backend.

Our approach is to device some mechanism using which design diagrams for web page can be created and represented in the text format. So that when user creates one web page design and save it, it generates one text file. The approach chosen is to represent the text per design element is as a XML tag. So every design element will have a unique tag as shown in Figure 1.

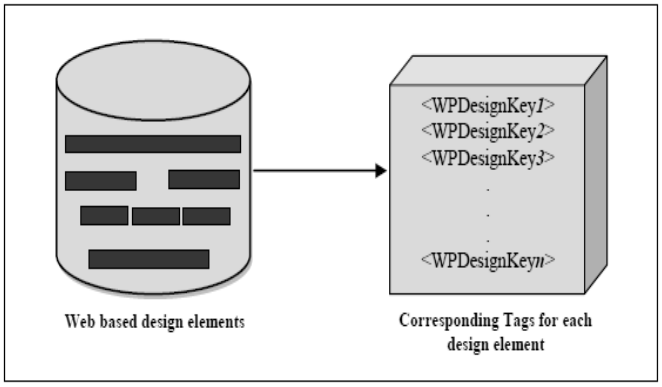

Figure 1: Association of Webpage Design Tag With Every Web Page Design Element

Initially the process start with making a new web page design document, which is XML, based. The system to create the web page design is having the collection of web page elements. All finite design elements have assigned names. User has to select and place these design elements while creating design document. When one web page design gets completed, there will be one complete XML document associated with it. These web page design repository will be a collection of associated XML document per design as shown in Figure 2.

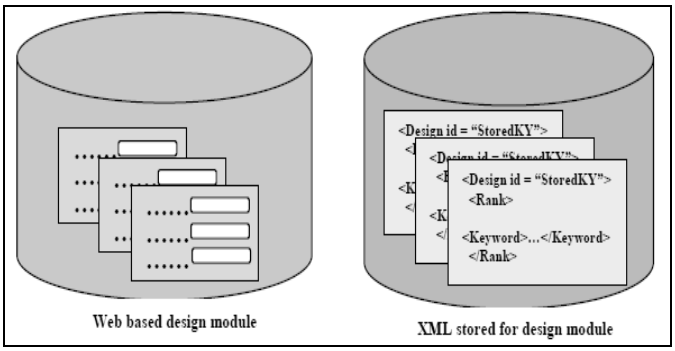

Figure 2: Each Design Module is Having Associated SDML Document

There will be a repository of designs after following this approach and every design is well defined in terms of XML tags. The text format of storing the design is having the predefined tags. Text format will be stored in XML notations. This is termed as Semantic Design Markup Language and is having the tags for all knows design elements.

When user picks one design element, its respective tag will be placed in its corresponding text file. When web page design complete for some scenario by following this approach, system will be having a complete representation in textual format in the form of tags.

The XML text file having the details of design figures is the basis of our research. The approach has many benefits. When pictorial representations are saved in text form, lots of possibilities are there which increase the efficiency of overall development process. Reusability of design search within the available web page design, will automatic generate design skeleton on the basis of requirement document, and available design text keywords for auto generation of test case scenarios.

Along with the design element tags, the XML-SDML file is having the information with the help of special tags (like <keywords>), which is important and is a key for further reusability criteria implementation and page rank number, which is used by Personal Agent and Service Agent for agent based web page semantic search. This additional information is the list of keywords that designer wants to assign to his design. Each web page design is having one name, keywords and some attributes.

Name identifies the design module, keywords are the handles for reusability, and attributes are factors, which design document gain after reusability. Agents review the web page design and assign ranks to each keyword in a web page. A high rank number by agent makes web page design a good candidate for reusability for particular type of user.

When some user wants to search the new web page based upon some keyword, he has to submit the keywords to be search to the Personal Agent. The keywords are analyzed and some keywords are evaluated by Personal Agent in terms of ranking. Next step is that agent looks into the available keywords of design module repository for the specific keyword.

The design repository will be stored for each design element in a centralized database of the Personal Agent. The design stored is of the atomic nature by representation of text format. Each elementary design element will be having one name and some associated keywords. These keywords will be reflected in the centralized repository.

Following figure 3 presents a graphical representation of the mechanism for web page design reusability that how initially web based design can be created from XML based design document and then from available design modules XML based designed can be created.

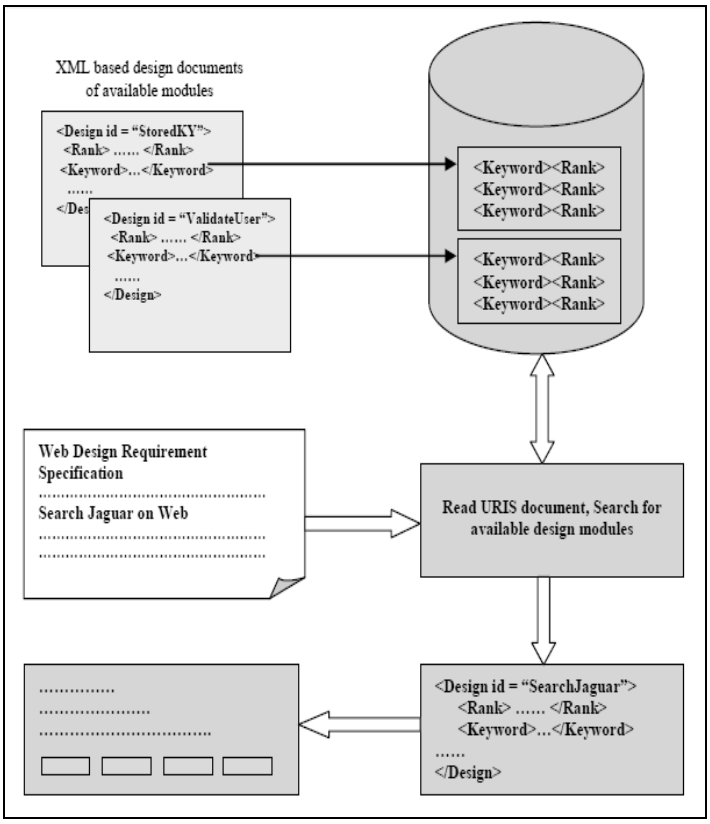

Figure 3: Mechanism for Web Page Design Reusability

In software development life cycle, when analysis phase is completed and the requirements are defined, next stage is to move to design process. The existing design repository will help out in reusing the design elements. Above figure shows the mechanism for web page design reusability. After following the approach for creating XML based web page design documents (SDML), there will be a database of XML sheets for each design module. Each design module is having the name and keywords associated with it.

<!ELEMENT *name>**

<!ELEMENT *keyword>**

Inference will be performed on the centralized design repository against the keywords from the Personal Agent. This will be a type of look-in search process in the design repository keywords of elementary design. When search gets completed, it will generate a minimum web page design document having the maximum reusability of existing modules. This SDML document is arranged in order to get the maximum requirement fulfillment for required web page design.

Also, the result of the inference on stored design repository will be having the elementary design modules having attributes also. These attributes specify the design reusability factor for a module (DRF). More the reusability factor of a module indicates that module is strong candidate for reusability in new design. Reusing a web page design module will increase the reusability count factor by one.

Design Storage and Retrieval

A. Create SDML based web page design and generate design repository per design module

Figure 4 is a logical representation while creation of new design. If user creates the new design and it is driven from some existing design element Dg..n, it will increment the design reusable factor DRF for Dg..n by one. This increment in DRF make makes design Dg..n a stronger candidate for further reusability

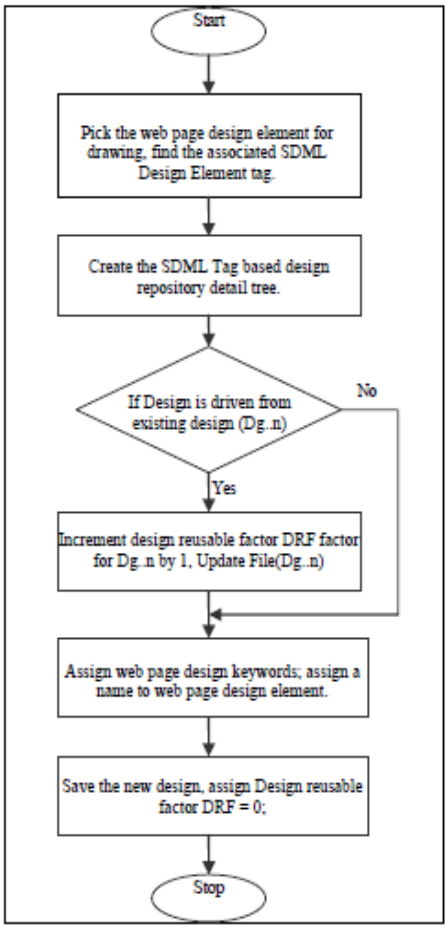

Figure 4: Create SDML Based Web Page Design

B. Retrieve SDML based design repository for reusability and create initial proposed design layout.

The following figure 5 shows how the DRF (Design Reusable Factor) helps in reusing the existing web page design. First the search for the keywords in existing design will be performed. The outcome then will be sorted in descending order on the bases in DRF. The top value of result outcome shows that first one in the search is the best candidate for reusability.

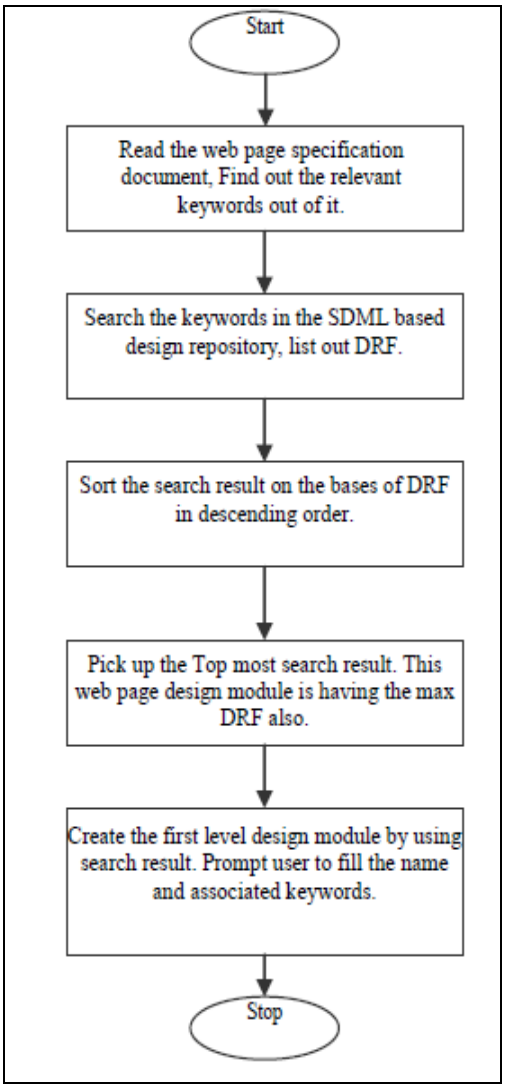

Figure: 5: Retrieve SDML based Design Repository for Web Page Design Reusability

EXPERIMENT SETUP

For the experimental validation of the proposed approach of web page design reusability and effort minimization, five keywords have been considered. These keywords are having maximum of six different sub keywords. Web page design requirement for these modules have been studied and one end user is considered for the experiment to which the finished design will be given for review. From requirement discussion to first draft, we record the time spend for each keyword. We also record the time consumed by the web page design architect. When this gets finished, we need to discuss it with the end user. Time record for the end user involvement was also recorded. We include the end user in the design phase for layout, look and feel type observation. We consider this as design prototype.

Result &Observations

Initial tables are the record of the efforts made by the web page design architect and end user for verification of the initial draft. This is the observation of the projects using conventional system and we will compare it with efforts consumed using our approach.

Table 1: Efforts (in Hrs) in Web Page Design Using Conventional Design Methods

Efforts in Hrs		
Keywords	**Conventional web page Design efforts**	**End User**
K1	15	7
K2	28	9
K3	16	6
K4	10	5
K5	19	6
K6	22	6

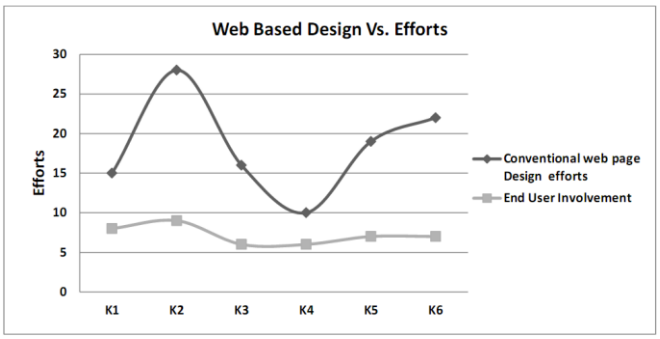

Figure: 6: Efforts (in Hrs) in Web Page Design Using Conventional Design Methods

The figure 6 represents that for a project of 6 modules, minimum time for design is 16 hours whereas maximum could be 37 hours for a project of 6 modules. This variation shows that module count is immaterial but the important is

nature of project. This observation when compared with the SDML based approach will shows very interesting results.

For our experiment with SDML based web page design, we are having 10 designs in the SDML storage and these are having 50% resemblance with the domain of the requirement. These design documents are having the keywords, which could be used during the search process. The SDML based design experiment for the reusability of existing design requires the framing of the requirement specification in some special format. So, we have made the requirement specification document. This is having the keyword-based text instead of the plain English based scenario discussions. Same projects with special formatting of the requirement specification are submitted to SDML inference engine.

This takes very less time in identifying the candidate from available stored design and displays the results. The time taken in framing the requirement and getting the first level design from existing designs is shown in the following table no. 2. It is approximately 1 hour because this is not simply selection process, user have to spend some time (in few minute) to identify which one is appropriate on the bases of some ranks, like reusability factor.

Table 2: Efforts (in Hrs) in Web Page Design Using SDML Based Web Page Design Methods

	Efforts in Hrs			
Keywords	Efforts in Framing Requirements	Effort in generating first level design	Agile client efforts	Total efforts in automation
K1	6	1	1	8
K2	9	1	2	12
K3	8	1	2	11
K4	5	1	1	7
K5	4	1	3	8
K6	8	1	2	11

Observations in Table 2 are showing the improvement in the efforts consumed. The only efforts consumed are in the framing of the requirement in the special format. The agile clients efforts, which can be a part of team also need to spend small time as compared to time spend by the user. We propose the involvement of the user in the design process with our mechanism. User interactions could be of the type of selecting one design layout out of available design layout produced by the system.

Figure 7 is the graphical representation of the table.

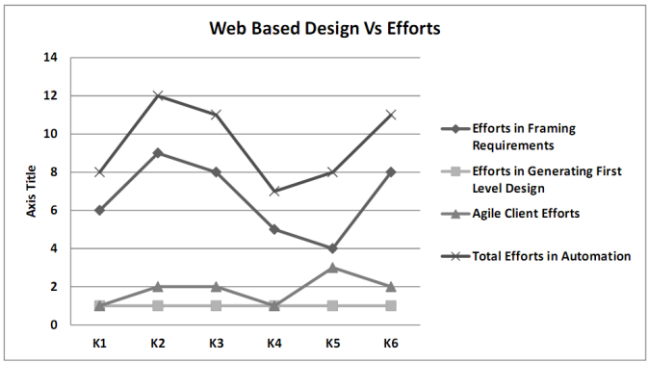

Figure 7: Efforts (in Hrs) in Web Page Design Using SDML Based Web Page Design Methods

The total time consumed in the finalization which is summation of time consumed in framing requirement, design generation by system and client time to select. We represent this as TED, i.e. total effort required for SDML based design.

$$TED = \sum RF_i + DG_i + ACE_i$$

RF is the efforts required for requirement framing for software project SP, DG is efforts in design generation and ACE is the agile client efforts.

As an observation from Table 3 we can see the total efforts consumed in the proposed design are less than that of conventional system which is showing the effort minimization after using the SDML based design approach and then applying the DNSIM (Design notation storage and inference mechanism) for reusing the same for achieving the qualitative designs.

Table 4.3: Total Efforts Comparison in Using Two Approaches

Keywords	Total efforts in SDML based approach	Total efforts in conventional web page design.
K1	8	23
K2	12	37
K3	11	22
K4	7	16
K5	8	26
K6	11	29

Figure 8 shows the graphical representation for the same.

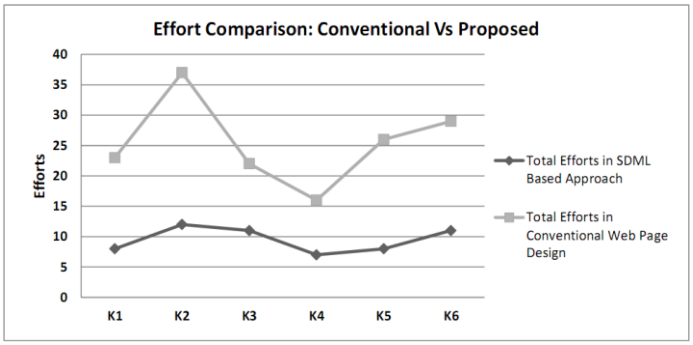

Figure 8: Total Efforts Comparison in Using Two Approaches

Figure 8 shows the comparison and figures in the total effort minimization. SDML based design approach uses requires the less effort, as compared to other methods. SDML based approach produces qualitative design, which helps in minimizing overall efforts in software development life cycle. The end user involvement and efforts for feedback are comparatively less in case of the SDML based approach.

For same projects taken in our experiment, we found that there is a big difference in the end user involvement. Table 4 and figure 9 present this difference.

Table 4: User Involvement Efforts Comparison in Using Two Approaches

Keywords	End User involvement (Conventional)	Agile client efforts
K1	8	1
K2	9	2
K3	6	2
K4	6	1
K5	7	3
K6	7	2

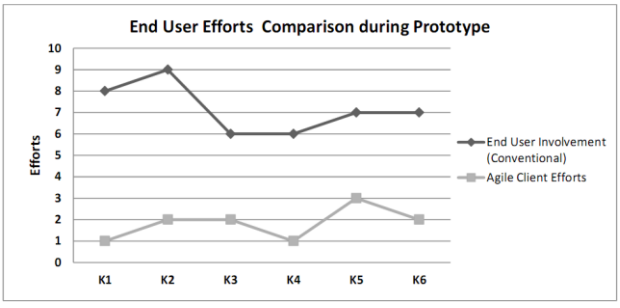

Figure 9: User Involvement Efforts Comparison in Using Two Approaches.

As for 1st project having 6 modules, the efforts consumed by the agile client for feedback are 6 hours for conventional approach where as its only 1 hour required for the SDML based user interface design. This also represents that it's a qualitative design approach towards effort minimization and making SDML less complex.

CONCLUSION

SDML based designs representation is a new approach for storing the design elements in the form of specially design SDML-XML tags. These are plain text files, so we can write programs to manipulate and analyze these files for further enhancement.

To simulate the effort minimization, we carry out above-mentioned experiment. It shows that the there is tremendous decrease in the user involvement. We carry out our experiment on five different projects having different number of modules. Table 1 is showing the efforts in hours and end user involvement using the conventional, approaches of software design engineering. The figure 6 depicts that there is big involvement of the end user in the design phase (we consider agile methodology where customer is a member of development team) as for 2nd project 28 hours efforts are from designer team and end user spend 8 hours. Table 2 shows the overall efforts for creating the first level design using SDML based approach. The total efforts in automation are less than that of the manual and conventional system. There is a significant minimization in the efforts of the agile client involvement as depicted from the Figure 7 and Table 2. Figure 8 shows the comparison between the total efforts of SDML based design and conventional design. In SDML based design, there is only on activity, which consumes time and that is the framing of the requirement specification document. This contains finding out the important keywords from the requirement specification document and using them later in the search process SDML based parsing. Figure 9 compares the involvement of the end user in the same projects using the two different approaches. It clearly shows that agile client involvement is reduced significantly after following the SDML based approach.

The web page design reusability leads to faster development of the application especially in the area where the web page is of prime importance. Big time and efforts, which are consumed in developing the core web page design functionality, could be minimized. The user involvement in the design phase and efforts could be minimized in the design process. This is because the design elements are stored in the textual format and we can apply several kinds of search algorithms for finding the best solution for the problem. Further, design storage in specified format makes the system scalable and

maintainable. The approach of the reusability of software design is novel.

BIBLIOGRAPHY

Aberer K. et al: (2004). Gridvine: Building internet-scale semantic overlay networks. Number 3298 in LNCS, pp. 107-121.

Arumugam, M., et al.: Towards peer-to-peer semantic web: A Distributed Environment for sharing Semantic knowledge on the web.

Bajwa I.S. and Chaudhary M.A. (2006). "A Language Engineering System for Graphical User Interface Design (LESGUID): A rule based approach: Information and Communication Technologies, 2006, 2nd volume, pp.3582-3586.

Bellifemine F., et al. (2003). JADE a white paper: EXP in search of innovation, 3.

Benjamins R. Decker S., Fensel D., Gomez-Perez A.: Building Ontologies for the Internet: A Mid Term Report, International Journal of Human Computer Studies (IJHCS), September, 687-712.

Bigus, J.P. et al. (2002). ABLE: A toolkit for building multiagent autonomic systems. IBM systems journal 41.

Castano, S. et al. (2003). Ontology-addressable contents in p2p networks. In Proc. Of WWW03, 14st SemPGRID Workshop.

Decker S., Erdmann M. (1999). Ontology Based Access to Distributed and Semi-Structured Information", Proceedings of DS-8. Kluwer Academic Publisher, Boston, pp. 351-369.

Doane S.M. and Lemke A.C. (1990). "Using cognitive simulation to develop user interface design principles", System Sciences, Proceedings of the Twenty-Third Annual Hawaii International Conference on Volume ii, pp. 547-554.

Doorenbos, R. B., Etzioni, O., and Weld, D. S. (1997). A scalable comparison shopping agent for the world-wide web. In Johnson, W. L. and Hayes-Roth, B., editors, Proceedings of the First International Conference on Autonomous Agents (Agents'97), pages 39–48, Marina del Rey, CA, USA. ACM Press.

Ehrig, M., Staab, S. (2004). QOM quick ontology mapping. Number 3298 in LNCS, pp. 683–697

Eichmann, D. (1992). Supporting multiple domains in a single reuse repository. InProc. Fourth International Conference of Software Engineering and Knowledge Engineering (SEKE'92), pages 164–169.

Fensel D., Ding Y., Omelayenko B., Schulten E., Botquin G., Brown M., and Flett A. (2001). "Product Data Integration in B2B E-commerce", IEEE Intelligent Systems (Special Issue on Intelligent E-Business), Vol. 16, No. 4, July/August 2001, pp. 54-59.

Fernández, M.; Gómez-Pérez, A.; Juristo, N. (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Symposium on Ontological Engineering of AAAI. Stanford (California).

Garland, A. and Alterman, R. (1995). Preparation of multi-agent knowledge for reuse. In Aha, D. W. and Ram, A., editors, Working Notes for the AAAI Symposium on Adaptation of Knowledge for Reuse, Cambridge, MA. AAAI.

Guarino, N. (1998). Formal Ontology and Information Systems'. In *Proceedings of Formal Ontology in Information Systems*, Netherlands: IOS Press.

Halevy, A., et al. (2003). Schema mediation in peer data management systems. In: Proceedings of the International Conference on Data Engineering (ICDE03), Bangalore, India.

Joachims, T., Freitag, D., and Mitchell, T. M. (1997). Web watcher: A tour guide for the world wide web. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), pages 770–777.

Jorge Cardoso, Martin Hepp, and Miltiadis Lytras (2008). "Real World Applications of Semantic Web Technology and Ontologies", Chapter-1.

Kuhnel, R. (1999). Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 104(1): pp. 1–69.

Lesser, V., Hornling, B., Klassner, F., Raja, A., Wagner, T., and Zhang, S. (2000). Big: An agent for resource-bounded information gathering and decision making. Artificial Intelligence, 118(1-2): pp. 197–244.

Marshall, C.C. and M. Shipman, F.M. (2003). Which Semantic Web?' In, *Proceedings of the fourteenth*

ACM Conference on Hypertext and Hypermedia. Nottingham, UK, New York: ACM Press.

Nejdl, W., et al. (2002). EDUTELLA: A p2p networking infrastructure based on rdf. In: Proceedings of the WWW2002, Honolulu, Hawaii, USA , pp. 604–615.

Quiroz J.; Shankar, A.; Dascalu, S.M.; Louis, S.J. (2007). "Software Environment for Research on Evolving User Interface Designs", Software Engineering Advances, 2007. ICSEA 2007. International Conference on 25-31 Aug. 2007, pp. 84-84.

Schelfthout K., Coninx T., Helleboogh A., Holvoet T., Steegmans E., and Weyns D. (2002). "*Agent Implementation Patterns",* Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies (Debenham, J. and Henderson-Sellers, B. and Jennings, N. and Odell, J., eds.), pp. 119-130.

Sichman, J. S. and Demazeau, Y. (1995). Exploiting social reasoning to deal with agency level inconsistency. In Proceedings of the First International Conference on MultiAgent Systems (ICMAS-95). AAAI Press/The MIT Press.

Tamma, V., et al. (2004). SERSE: searching for semantic web content. In: Proceedings of ECAI .

Terry R. Payne, Rahul Singh and Katia Sycara (2002). "Calendar Agents on the Semantic Web", IEEE Intelligent Systems Volume 17(3), pp. 84-86.

Tim Finin, Rich Fritzson, and Don McKay (1992). "A Knowledge Query and Manipulation Language for Intelligent Agent Interoperability", Fourth National Symposium on Concurrent Engineering, CE & CALS Conference, Washington DC.

Turner, J. and Jennings, N. (2000). Improving the scalability of multi-agent systems. In Wagner, T. and Rana, O., editors, Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, volume 1887 of Lecture Notes in Artificial Intelligence, pages 246–262. Springer Verlag.

Wooldridge, M., Bussmann, S., Klosterberg, M. (1996).. "*Production sequencing as negotiation*". In: Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-agent Systems, London.

Wooldridge, M., Jennings, N. (1995). Intelligent agents: Theory and practice. Knowledge engineering review 10, pp. 115–152.

Zilberstein, S. and Russell, S. (1995). Approximate reasoning using anytime algorithms. In Natarajan, S., editor, Imprecise and Approximate Computation. Kluwer Academic Publishers.

Corresponding Author

Herald Noronha*

Research Scholar, CMJ University, Shillong

E-Mail –