IPHV7I3013X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

SEMANTIC MODULARITY IN ONTOLOGY'S AND INTEGRITY CONSTRAINTS

[#1]A. MALLAREDDY, **[#2]ESKILLA SATHISH**
Research Scholar (JNTUH), Professor & HOD (CSE) **M.Tech (CSE) Pursuing**
Department of Computer Science & Engineering,
Sri Indu Institute of Engineering & Technology, Sheriguda(M), Ibrahimpatnam (M), RR Dt. Hyderabad – 501510.

Abstract : A good practice is therefore to build on the efforts made to design reference DMSs whenever we have to develop our own DMS with specific needs. A way to do this is to extract from the reference DMS the piece of schema relevant to our application needs, possibly to personalize it with extra-constraints with respect to our application under construction, and then to manage our own dataset using the resulting schema. Our proposed approach is plan to evaluate our approach, in particular to compare the size of the modules extracted by our algorithm to the results provided by, We also plan to apply our algorithms to the real usecase of the My Corporis Fabrica DMS, mentioned in the introduction, which has been developed manually as a personalization of the (reference) Foundational Model of Anatomy DMS. Finally, we plan to extend our approach to distributed module-based DMSs, where answering queries combines knowledge of several modules associated with possibly several reference DMSs. We revisit the reuse of a reference ontology-based DMS in order to build a new DMS with specific needs. We go one step further by not only considering the design of a module-based DMS (i.e., how to extract a module from a ontological schema): we also study how a module-based DMS can benefit from the reference DMS to enhance its own data management skills. We carry out our investigations in the setting of DL-lite, which is the foundation of the QL profile of OWL2 recommended by the W3C for efficiently managing large RDF datasets. RDF is the W3C's Semantic Web data model, which is rapidly spreading in more and more applications, and can be seen as a simple relational model restricted to unary and binary predicates. In addition, DL-lite comes with efficient inference algorithms for querying RDF data through (DL-lite) ontologies and for checking data consistency with respect to integrity constraints expressed in DL-lite. Our contribution is to introduce and study novel properties of robustness for modules that provide means for checking easily that a robust module-based DMS evolves safely with respect to both the schema and the data of the reference DMS.

Keywords: *Novel based property, Rule based reff. Data management systems, Ontology.*

I. INTRODUCTION

The amount of web-based information available has increased dramatically. How to gather useful information from the web has become a challenging issue for users. Current web information gathering systems attempt to satisfy user requirements by capturing their information needs. For this purpose, user profiles are created for user background knowledge description .User profiles represent the concept models possessed by users when gathering web information. A concept model is implicitly possessed by users and is generated from their background knowledge. While this concept model cannot be proven in laboratories, many web oncologists have observed it in user behavior. When users read through a document, they can easily determine whether or not it is of their interest or relevance to them, a judgment that arises from their implicit concept models. If a user's concept model can be simulated, then a superior representation of user profiles can be built. To simulate user concept models, ontologies—a knowledge description and formalization model—are utilized in personalized web information gathering. Such ontology's are called ontological user profiles or personalized ontologies. To represent user profiles, many researchers have attempted to discover user background knowledge through global or local analysis. Global analysis uses existing global knowledge bases for user background knowledge representation. Commonly used knowledge bases include generic ontology's (e.g.,WordNet), thesauruses (e.g., digital libraries), and online knowledge bases (e.g., online categorizations and Wikipedia). The global analysis techniques produce effective Performance for user background knowledge extraction.

II.RELATED WORK

2.1 Document Retrieval

The concept "text retrieval" is used today to describe a certain computerized method for retrieving documents. The process itself is quite general and often used in everyday situations. One does, for instance, suspect that one has caught

ISSN Online: 2319 – 9253
Print: 2319 – 9245

a certain germ which has been reported in the news lately. The most usual way to find the news items would probably be to browse through newspapers until hitting the relevant articles. Reading with sufficient care, one would obtain a perfect result in the sense that all relevant articles are being retrieved.

This manual retrieval does not offer problems as long as the number of documents (in our example, the stack of newspapers) is small. Confronted with all the newspapers of a country, however, the situation would be different. It would indeed be doubtful whether one would consider the task at all. In such cases computerized retrieval would be advantageous.

2.2 Full Text Retrieval

The term "full text retrieval" is not used in this book as it is ambiguous. It may either denote that the source is represented in authentic form, or that the principle of text retrieval is employed, regardless of the document design. In this book, text retrieval is the name only of a certain retrieval method and implies nothing of the document design.

Text retrieval does not presume that the source is edited before being converted to machine readable form. It is usual to represent the sources in a general text format. In such cases, one often characterizes the method as full text retrieval. In the input phase, it is in principle sufficient that "end of document" and "end of word" is defined. In some systems, also "end of sentence" and "end of paragraph" may be defined, allowing retrieval also on the distance between words in terms of these text segments.

Retrieval is based on word occurrences and combinations, and in a text retrieval system any word in the material may be retrieved. In order to obtain a high degree of efficiency, one does however exclude some common words or stop words like prepositions, conjunctions, articles, pronouns, certain verbs, etc occurring frequently in the text and which are rarely helpful in describing the content of a document with respect to the retrieval system. Such stop words generally make up some 35-45 per cent of the text volume.

2.3 Interest Retrieval

Text retrieval systems are well suited for interest retrieval. The documents are stored in natural language, making it possible for the user to employ any word of the text in the search request. For specific problems, specific requests may be constructed; and properties of natural language documents make it possible to rank retrieved documents, for instance using word statistics.

They are, however, less suited for fact retrieval. The system has no exact data on the content of the documents (only on which words they contain), and cannot respond to requests presuming such data. The system may furnish information on which words the documents contain and in what sequence, and if this is sufficient for the fact retrieval request, it may serve such a purpose. Retrieving on a compilation of statutes, it may for instance specify how many documents are citing a certain statute by using the identification of this statute as a search request.

Though free text formats do not presume editing, inclusion of fixed fields containing predefined data (name of author, title, date, citations, etc), editing is in practice often implied. The actual computer system has facilities to exploit such fields as well. In this chapter we shall, however, concentrate on the simple, unedited form in which only a raw text is inputted. But systems having a superstructure of fixed fields have the possibility of being successfully used both for fact and interest retrieval.

Existing Method:

The current trend for building an ontology-based data management system (DMS) is to capitalize on efforts made to design a preexisting well-established DMS (a reference system). The method amounts to extracting from the reference DMS a piece of schema relevant to the new application needs – a module –, possibly personalizing it with extra-constraints with respect to the application under construction, and then managing a dataset using the resulting schema.

III. RETRIEVAL PROCESS

The retrieval process is defined as the total process from the emerging problem (information need) to the final answer from the system.

The first process (the transformation of the problem to a request) is the responsibility of the user. In a certain search language, the user attempts to express his information need according to the requirements of the system. This is generally the only possible way in which the user is permitted to specify his problem. It is therefore important that the search language is flexible, and is exploited in such a way that the request reflects the problem.

The second subprocess is carried out by the retrieval system. Based on the request, the system attempts to find all and only the relevant documents. The user may, for instance, in browsing through the retrieved documents, get ideas for improving the search requests.

IPHV7I3013X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

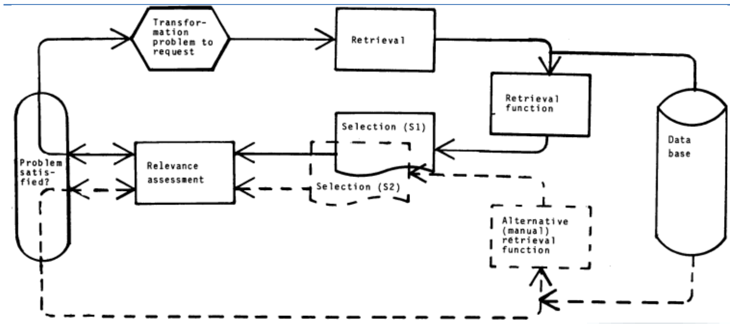

Fig: process of Retrieval Process

Assessing the result generally implies a comparison of the result to the set of relevant documents the user would have identified in reading the total number of documents in the data base. Naming this set S1, and the retrieved documents S2, the intersection between the two sets will be a measure of the quality of the computerized search.

Existing System: The current trend for building an ontology-based data management system (DMS) is to capitalize on efforts made to design a preexisting well-established DMS (a reference system). The method amounts to extracting from the reference DMS a piece of schema relevant to the new application needs – a module –, possibly personalizing it with extra-constraints w.r.t. the application under construction, and then managing a dataset using the resulting schema..

Problems on existing system: This is Method Not Maintain Easy.

Proposed System: Here, we extend the existing definitions of modules and we introduce novel properties of robustness that provide means for checking easily that a robust module-based DMS evolves safely w.r.t. both the schema and the data of the reference DMS. We carry out our investigations in the setting of description logics which underlie modern ontology languages, like RDFS, OWL, and OWL2 from W3C. Notably, we focus on the DL-liteA dialect of the DL-lite family, which encompasses the foundations of the QL profile of OWL2 (i.e., DL-liteR): the W3C recommendation for efficiently managing large datasets.

Advantages: This is very useful to maintain Data, Search and retrieve the data is very Easy.

IV. EVALUATING PROVIDED PRINCIPLES

For evaluating and testing the provided principles which are shown in table 2, we proposed an architecture which was named MEDA. MEDA which is shown in figure 1 has been provided based on modular principles in table 2. In this architecture to reach maximum quantity of modularity and based on M2, each module has to perform all its functionality by itself and Core just run final instructions that have been sent by each module.

To reach maximum quantity of extendibility and modifiability, we need a dynamic structure and dynamic channel to communicate between modules. So, extensible Markup Language (XML) technologies [12] have been used to have dynamic structure and communication protocol between modules. To have dynamic channel, a combination of Event Driven Architecture [11] and a simple type of Bus Architecture [13] has been used. Event-Bus module in Core does these functionalities. To support M1 and M4 from table 1, CAL (Content Access Layer) as a content layer has been placed in MEDA. CAL must execute instructions related to content access. Instruction is sent from modules to Event-Bus and then from Event-Bus to CAL. Contents can be Data, File or Web Services. In CAL, we need modular content access that modules can't access content of each other. For example in modular file access we must use operation system APIs that modules can't access files of each other. In modular data access we need assign a specific database username and password to each module that can't access table of each other.

Module Installer Layer must install and hold each modules and their profile. RTL (Run Time Layer) must do all runtime actions except content access instruction. Exception Manager must control and manage all exceptions that are sent from modules. AA Manager do authentication of whole system and based on M2, M4 authorization of system delivered to each module. Module Presentation must control presentation actions. Theme changing and language changing are done by Module Presentation. For example when a user changes the theme of CMS, theme changing as an event is sent to all modules by Module Presentation. Systems Decoration must manage template of CMS and locate modules presentation part in considered place.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

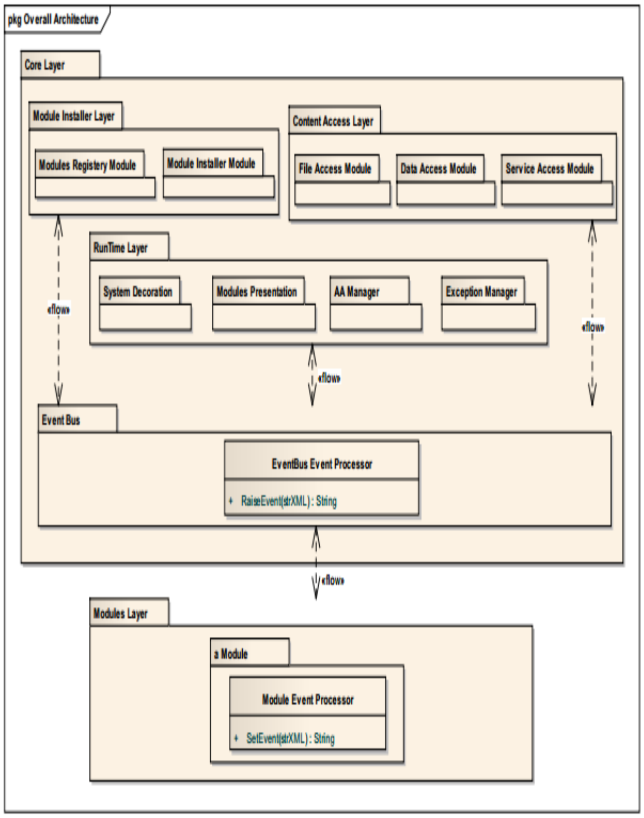

Figure 1: MEDA, provided architecture for evaluating new principles.

V. OVERALL SEQUENCE OF EVENT RAISING IN MEDA

In MEDA, each module and Core as a module, have to implement a class by the name of Event-Processor and use Event-Bus as a channel for interacting messages between Core and modules. In this architecture we put messages in the form of Events. All of modules and even the Core use Raise Event method from Event-Bus for sending events and Event-Bus uses Set Event method from Event-Processor (implemented in each module) for sending delivered events to target modules. Interaction between Core and modules are prepared by two types of XML files by the name of Document Type Definition 1 (DTD1) and DTD2. For example after a user logged in to system, Core will send a request (in the form of an event) to get User Access List from all the modules to represent User Control Panel. This action will perform by the use of an event like: +get Access List(String Username):String; DTD1 is a template for sent events. DTD1 usually contains event-type, event-name, input parameters names and values, return type and value, event-sender, event-receiver(s), etc. But while developing each module of MEDA, we must define standard DTD for module event list. Each module for sending an event must put it in the form of DTD1 and invoke Raise Event method from Event-Bus. Then Event-Bus analyze delivered event and in order to sending event to target modules, use the Set Event method.

VI.CONCLUSION

In this paper, new decentralized principles for modular systems have been provided. These principles distribute Core functionalities between modules based on robust object oriented thinking. So, dependencies between modules decrease saliently and existing systems turn to more modular systems. Therefore module development will take extra effort than before. Finally we conclude that the data management with rule based mechanism are shown effective results. Usage of ontology are analyzed and implemented effectively. Data management system uses the reference models to handle the data. In this study we focuses on reference systems with various data management systems by using ontology and referencing done though rule based mechanism by using novel properties.

REFERENCES

[1] De-Nian Yang, Member, IEEE, and Ming-Syan Chen, Fellow, IEEE,"Efficient Resource Allocation for Wireless Multicast", IEEE Trans.Mobile computing, vol.7, no.4, Apr.2008.

[2]Unlicensed Mobile Access (UMA), http://www.umatechnology.org/index.htm, 2007.

[3] D. Waitzman, C. Partridge, and S. Deering, Distance Vector Multicast Routing Protocol, IETF RFC 1075, 1988.

[4] J. Moy, Multicast Extensions to OSPF, IETF RFC 1584, 1994.

[5] D. Estrin et al., Protocol-Independent Multicast-Sparse Mode (PIMSM): Protocol Specification, IETF RFC 2117, 1997.

[6] A. Ballardie, Core-Based Trees (CBT Version 2) Multicast Routing Protocol Specification, IETF RFC 2189, 1997.

[7] S. Bhattacharyya, An Overview of Source-Specific Multicast (SSM), IETF RFC 3569, 2003.

[8] T.G. Harrison, C.L. Williamson, W.L. Mackrell, and R.B. Bunt, "Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts," Proc. ACM MobiCom, pp. 151-160, 1997.

[9] C.R. Lin and K.-M. Wang, "Mobile Multicast Support in IP Networks," Proc. IEEE INFOCOM, vol. 3, pp. 1664-1672, 2000.

[10] Y. Wang and W. Chen, "Supporting IP Multicast for Mobile Hosts," ACM Mobile Networks and Applications, vol. 6, no. 1, pp. 57-66, Jan. 2001.

[11] J.-R. Lai, W. Liao, M.-Y. Jiang, and C.-A. Ke, "Mobile Multicast with Routing Optimization for Recipient Mobility," Proc. IEEE Int'l Conf. Comm. (ICC '01), vol. 5, pp. 1340-1344, 2001.