Technology Enablers for Building Content Management Systems

Vasudeva Varma

Abstract

Managing content in a reusable and effective manner is becoming increasingly important in knowledge centric organizations as the amount of content generated, both text based and rich media, is growing exponentially. Creating content is expensive and unless it is staged, deployed and reused effectively, these costs cannot be justified. An important aspect of content management technologies is that they are far from being mature and they are getting better with time and this will continue to stay this way for some more time. In this paper we discuss important features of next generation content management systems and key technology enablers that make it possible to achieve next generation functionality. We describe two important technology enablers that are implemented with state of the art research besides identifying evolving and futuristic research areas that will help push the content and information systems to the next level. We describe a technology framework for managing a unified taxonomy and ontology network and a common messaging platform.

Keywords : Content Management Systems, Content Management, Common Message Platform

0. Introduction

We have entered an era of information overload. This is in contrast to what we have experienced in the past century where the main challenge was to find enough information. Most organizations are transforming into knowledge organizations where the key assets of organization are turning out to be people and knowledge. Every knowledge centric organization is producing more and more content and information. The new generation information and content management challenges can be classified into two major groups: information staging and information retrieval. Information staging include activities such as finding the information sources, building content crawlers, building content indexers, metadata creation and building tools to characterize the content. Information retrieval phase includes content query processing, content delivery using push and pull technologies, content monitoring and feedback engines.

As the information, documents of structured and unstructured nature are growing exponentially; we have a challenge of finding most relevant document(s) in the least possible time. Hence, obtaining very high precision and recall in information retrieval systems is very important. In addition, mergers and acquisitions are major hurdles faced by the twenty first century content management system architects. As number of organizations are being merged or acquired, making sure that the content of organizations can also be merged seamlessly is also very important. We need to plan for open architectures while building content and information systems to enable communication between completely different systems.

Content management is a discipline that manages timely, accurate, collaborative, iterative, and reproducible development of web and inter-organizational digital assets. It combines a mechanism to store a collection of digital assets with processes that seamlessly mesh the activities of people and machines within an organization. Content management responds to the unique combination of problems posed by digital asset development, typically web related.

1. Content Management System Features

Content management has to support various activities – creation of content, content publishing, storage and efficient retrieval. The structure of the content, content types and content aggregation must be addressed in an effective and unified platform.

- Structure of the Content: The structure of the document itself contains a lot of useful information and provides useful semantic information. This information will be used in the content analysis stage. For example, knowledge about a particular section like "summary" in a document solves half the problem – because we know what to expect. The retrieval and extraction efficiency are dramatically improved if we have the structural knowledge of the document.
- Content Types and Content Sources: The right technology should enable parsing, extracting and indexing various content types and content sources. Content Types include – text, voice, video and graphics and non-trivial or unstructured textual information. Content sources can be configured by the client application – for example, email documents, company information databases, websites, HTML, Word, PDF documents, digital video libraries etc. Each of these content types and content sources are configurable by the client application.
- Content Aggregation: We need to deal with multiple rich media content types. What is stored electronically is only a document or a database record. When presented finally to the user, the information becomes and turns into Content. For example, a single search may result in various rich media objects that are interlinked or completely independent from each other. We must provide powerful presentation and content aggregation schemes so that the retrieval of information is unified and can be presented through multiple devices. The application program can combine the search results in a meaningful manner and present the rich-media objects using various devices.

Enterprise Content Management (ECM) represents various technologies for web content management, document and record management and digital asset management [PWC, 2003]. Web content management deals with the process of content creation, revision and approval, and a version control system. A workflow system will enable all these processes in an appropriate sequence. Document management systems provide role based access to individuals, collection, meta-tagging, and coordination for creation, modification, use and storage of electronic and paper documents through their life cycle. Document management systems support versioning as well as methods of storing essential metadata so that the information can be classified, searched and reused more effectively. Content integration technologies provide opportunities to feed relevant and approved content into key business applications.

Enterprise Content management can also be seen as an amalgamation of related product categories. Various software vendors and service providers were catering the needs of web content management, documents and records management, search engines, portals, Knowledge management tools, syndication of content independently and later realized that all these individual product and service markets can be brought under a single umbrella of content management. Digital asset management and digital rights management are two other functional solutions that are related to the ECM.

1.1 Content Management Systems - Future Trends

IDC analyst, Susan Feldman [Feldman, 2001] observed that the next major stop in content management, Search and information retrieval domain is "conversational systems" where the user and the application are engaged in a dialogue to arrive at the right content within a shortest time possible. This area is making use of various technologies including advanced natural language processing, semantic analysis,

machine learning, and user modeling. She also felt that the growth rate of rich media content will be more than text based content.

A very important trend one can notice in the market today is about personalization and localization of search and content. Search results become more meaningful if they are customized to the user who is making the query and address his or her information need in a given context.

Given the changing landscape of technology trends, demands of knowledge centric organization and the criticality of building effective content and information management system, we need to build an open ended system that can accommodate the changes in technology, user requirements as well as platform dependent specifications. In this paper we discuss important technology drivers that will shape up future content management systems and give a component based framework as a common messaging platform for building content management systems. We believe this framework can help in coming up with large scale, flexible and scalable content and information management systems.

2. Key Features of Next Generation Content Management Systems

In this section we describe some of the key features and functionality of an effective content management system. These features can guide in identifying the technology drivers that will enable us build the next generation content management systems.

2.1 Enhanced Content Crawling

It is important to capture the digital assets from various sources and systems including that of file storage, web, databases and storage networks. The multi-media content can originate from any sources. The content crawlers and document filters can automatically 'grab' content from content databases, video sources, voice sources, XML Sources, emails etc. The content capture is shown in the following diagram:

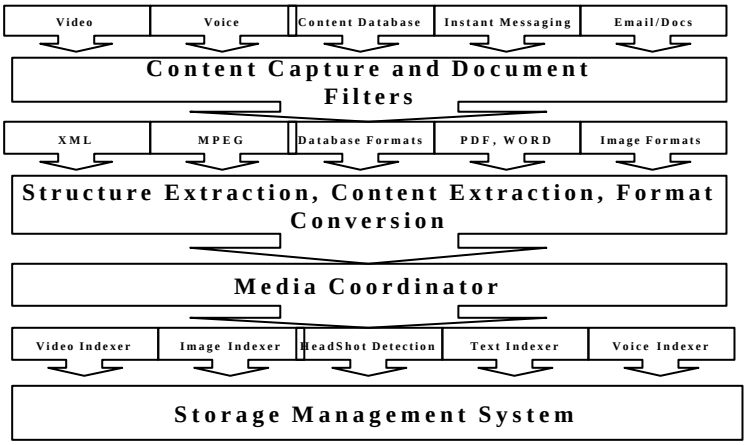

Figure 1: Content Crawling and Object/Document Management in Content Management Systems

The content from various sources is being captured by crawlers. Depending on the document type and document format, various filters can be used to separate the media specific content.

2.2 Digital Document and Object Management

Digital document and object management functionality deals with storage, filtering, extracting, structural analysis of rich media documents and objects. Consider for example, a multi media document that consists of a lot of structural information, textual information and images. We need to be able to parse and extract the structural information and different media objects from the document. Once this is done, structural analysis of the document provides us a lot of useful semantic information about the document. A format conversion might take place (for example, converting word file to ASCI text file) and is handed over to the media coordinator. The media coordinator will in turn pass the content to specific media indexing routines. We need store the structure of the document, pass the different media objects to their respective indexers and maintain the document in its original form in centralized storage servers. We need to create unique document identifiers at the same time. After indexing the content, storage management system will take over to capture both the document and the index of the document in appropriate database structures.

2.3 Storage Management

The back-end support issues for a multimedia content publishing system are very complex and need careful planning and implementation. This typically involves not only the storage management but also storage tracking, as we cannot expect entire content to reside on just one server. We need to provide tools to manage the digital assets in terms of their storage and manipulation that include re-packaging, re-creation and re-structuring

2.4 Media Specific Categorization and Indexing

Once the document is parsed and various media objects are extracted from the document, we need to index the sub-documents (or media objects) based on the media type. For example, video objects need to be parsed and indexed by the video indexers, similarly, textual objects need to be indexed by text indexers, image objects need to be indexed by the image indexers. There may be specialized and more than one indexer for any specific media type. The content management system architecture needs to be extensible to add new indexing engines.

2.5 Document Retrieval

It is important to find the right digital content in the shortest interaction time and in a very intuitive manner. We need to employ techniques such as “pearl growing” (improving and building upon the initial query). Ability to combine keyword or text based approach with sample image or image parameter based approach. An example query would look some thing like: “show me all the Toyotas which are shaped like this [insert or select an image] and are in black color and registered in Delhi”.

The system should be able to navigate through vast digital content with ease and efficiency. Users should be able to interact with the digital content servers in a very meaningful manner using the latest technologies such as conversational systems to arrive at right content in fastest possible manner.

2.6 Summarization

Summarization is important for textual documents as well as rich media documents. For example, video summarization is possible using the techniques that include video skimming or fast flipping of select frames and video information collages. For audio and text media we will use the summarization techniques developed in natural language processing and linguistics research. For images we can create thumb nails.

2.7 Personalization

Customization of the content for an individual requires mixing and matching content elements. Personalization has become very important in web content management systems and this area has proven to be highly specialized and complex. A new trend of personalization of results obtained by search engines is gaining popularity within search community. Personalization takes into account various parameters such as past and present user behavior, the context in which the search is being made, and predicted information need of the user.

3. Technology Enablers for Content Management Systems

It is not easy to build a high end content management system without certain technology enablers also known as technology drivers. To achieve the functionality described in the previous section, we need to build the foundation in the form of technology enablers. These technology enablers include good linguistic analysis, rich media processing, a common messaging platform where various components of the content management systems will be able to communicate in an optimal and uniform fashion and lastly a network of ontologies and taxonomies that form backbone of content processing. Linguistic analysis and rich media processing are key research areas that hold the key to the success of future content and information management systems. These research areas are growing very fast but are still far from being mature. In this section we deal with the other two technology enablers, namely common messaging platforms and taxonomy and ontology networks. We share our experience of creating these two fundamental building blocks in the context of architecting a framework for content management and knowledge management systems.

3.1 Universal Taxonomy and Ontology Network - UTON

The main purpose of any ontology is to enable communication between computer systems in a way that is independent of the individual system technologies, information architectures and application domain. The key ingredients that make up ontology are a vocabulary of basic terms and a precise specification of what those terms mean.

The term ‘ontology’ has been used in this way for a number of years by the artificial intelligence and knowledge representation community, but is now becoming part of the standard terminology of a much wider community including object modeling and XML.

Adoptable, high performing, large scale ontologies that can be extended to support multi-media play a crucial role in building effective content and information management systems and applications. This section describes the architecture of Unified Taxonomy and Ontology Network (UTON).

The ontology or taxonomy defines the central semantic network – in other words, it is a repository (industry specific, customizable or universal) that servers as basis for all the indexers. The content management system should be able to operate with multiple taxonomies and ontologies at the same time. It should be possible to switch between taxonomies or ontologies depending on the context and the input document. Hence it is important to come up with a framework where multiple taxonomies or ontologies can co-exist and accessed using unified protocols.

The content management systems and information management systems make use of ontologies at several functional points that include: document categorization, indexing, document (parts or entire document) retrieval, user query expansion, query matching, and result verification. Since rich media documents are also becoming pervasive and important (perhaps more important than the textual

documents) there is an emphasis on extending the ontologies work for multimedia documents as well. For this purpose, we need to build ontologies that support rich media document processing.

Ontologies can be used to provide semantic annotations or meta-tagging for collections of images, audio, or other non-textual objects. These annotations can support both indexing and search. Since different people can describe these non-textual objects in different ways, it is important that the search facilities go beyond simple keyword matching. Ideally, the ontologies would capture additional knowledge about the domain that can be used to improve retrieval of images.

UTON stores multi media concepts, relations among these concepts, cross linkages, language dependencies in its repository and provides interfaces to storage and retrieval functionality and the administrative functionality (including user and version management). The knowledge and semantic information is stored within the network in the form of a DAG (Directed Acyclic Graph). The storage and retrieval interfaces provided by ontology network are being used by various media indexing and categorization components. Ontology developers, editors and administrators will have different interfaces.

All these interfaces interact with higher level UTON objects such as Ontology, Concept, term and relation. If ontology consists of concepts belonging to more than one domain or sub domains, then another higher level object called context will come into play to help disambiguate concepts belonging to more than one domain. The following paragraphs describe each of these higher level objects:

- Ontology: the ontology is the topmost entity, necessary because the intention of UTON is to contain a network of taxonomies and ontologies, likely to be contributed by different sources. Depending on the number of domains the ontology contains a set of contexts will form the ontology itself. As attributes, the ontology has a name (mandatory and unique), a contributor, an owner, a status ("under development", "finished" ...) and documentation (an arbitrary string in which the contributor or the owner can specify relevant information).
- Context: a context is actually a grouping entity; it is used to group terms and relations in the ontology. Within a given ontology, every context should have a unique name. The context object comes into picture when there is a possible existence of ambiguous concepts (see below for the description of concept), terms and relations among them when a given ontology covers more than one domain or sub domain, which is typically the case.
- Concept: a concept is an entity representing some "thing", the actual entity in the real world and can be thought as a node within the ontology structure. Every concept has a unique id. A concept also has a triple "source-key-value", which is the description for that concept. The "source" identifies the source from which the description originates, the "key" is a string which gives a hint to the user on how he should interpret the value, and finally the "value" is the description of the concept. One concept can have more than one source-key-value triple, and thus have its meaning described in different ways. As an example, let's consider WordNet [Fellbaum, 1999]. In WordNet synsets denote a set of terms (with their "senses") which are equivalent. Every term also has a glossary, which is an informal description of the meaning for that (particular sense of the) term. In this respect, from WordNet, we can extract two different descriptions for a concept, two different source-key-value triples, namely the glossary (Source: WordNet - Key: Glossary - Value: "<informal description denoted as a glossary in WordNet>") and the synset (Source: WordNet - Key: Glossary - Value: <enumeration of synonyms forming the synset>). As a different example, when a concept exists in various media (text, video, audio and image), a concept represented using source-key-value triple will give the appropriate media value, when retrieved using appropriate key.

- Term: a term is an entity representing a lexical (textual) representation of a concept. Within one context, a term is unambiguous and, consequently, it can only be associated with one concept and of course, several different terms within one context can refer to the same concept, implicitly defining these terms as synonyms for this context. Terms in different context can also refer to the same concept, and in this way implicitly establish a connection between these two contexts.

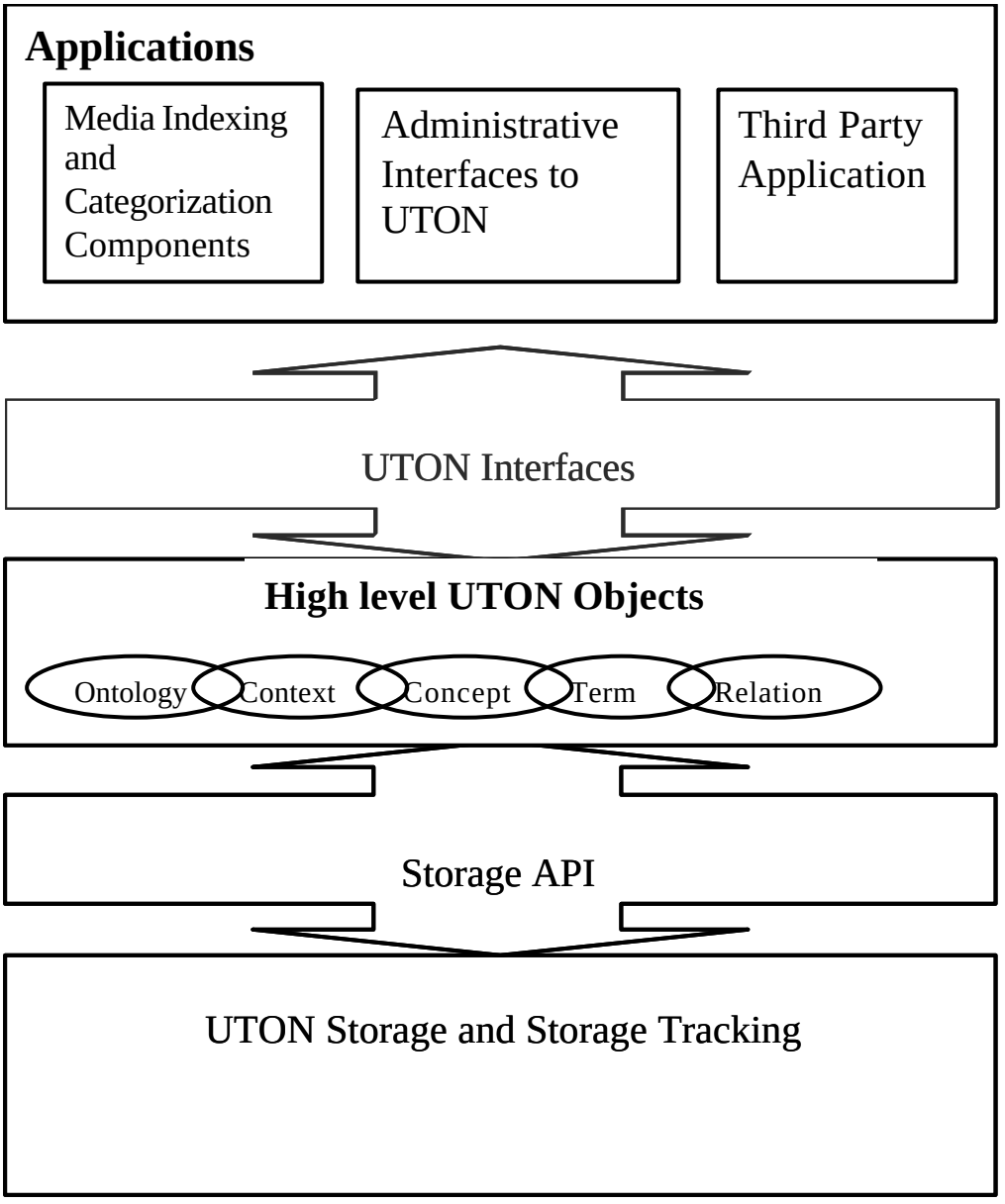

Figure 2: Architecture of UTON

- Relation: a relation is a grouping element; it can be interpreted as a set of triples consisting of a starting term (also called the "headword" of the relation), a role (relation name) and a second term (also called the "tail" of the relation).

As we can see in the above figure, the general architecture components are:

- UTON Storage: the storage system is the place where the UTON data is stored – typically a Relational Database Management System (RDBMS).
- Storage API: Provides a unified access to the basic structures of UTON. The API should be accessible from any high level programming language.
- Higher level UTON objects: UTON objects are expressed in a data description language format, or as objects in any high level programming language. They are retrieved and stored using the storage API.
- Applications: applications can use the UTON by integrating the ontology objects returned from the storage API in their program code.

This architecture and design of UTON [Varma, 2002] will enable multiple ontologies and taxonomies to co-exist and make it possible to access them in a unified manner. Our major focus is to build a network of large scale ontologies and taxonomies that are highly scalable and with high performance and guaranteed quality of service. All the components can be distributed and can be running on a set of server forms to obtain the required scalability and performance.

We have developed UTON in the context of developing information extraction, indexing and categorization engines for a content management system that is heavily rich media oriented. WordNet played a major role in coming up with an initial ontology.

3.2 An Approach to Developing Common Messaging Platform

A major challenge in building content management system is in scaling up the system. In real world scenario, we have huge amount of content flowing in from various sources and each content document may be separated into sub documents depending on the type of media. These subdocuments need to be indexed and characterized by appropriate indexer. In addition, metadata like named entities, author information is extracted and summaries are generated and documents are categorized into predefined categories. This new metadata has to be seamlessly merged into the characterization of existing document repository. All this has to be done in real time to make sure that users have access to latest information and content.

Each of the tasks mentioned above may be executed by a farm of severs where the appropriate components are deployed. These components need to be supported by an efficient communication channel that does not become a bottleneck in achieving high performance. If each component is allowed to talk to every other component then network traffic can become very high and congestion can occur causing the content management system to breakdown. To address this important performance engineering issue, we have designed a common messaging platform where every message from every component will be sent through this platform and will be received by the appropriate target component. The solution is building content based common messaging platform where every message is very thin (textual – XML based) but will enable the target component to take act on the data. We were scouting for commercial off the shelf components and selected "Elvin" which we later customized for our need.

Our approach to building the common messaging platform is based on component architecture. All modules are components and these components are connected by a messaging platform. Each of these components can have multiple instances and these instances can be running on multiple machines.

Elvin is a messaging platform that keeps track of the state of the components that are registered with it and receives and passes the messages appropriately to the corresponding components. We used the messaging platform in the following manner.

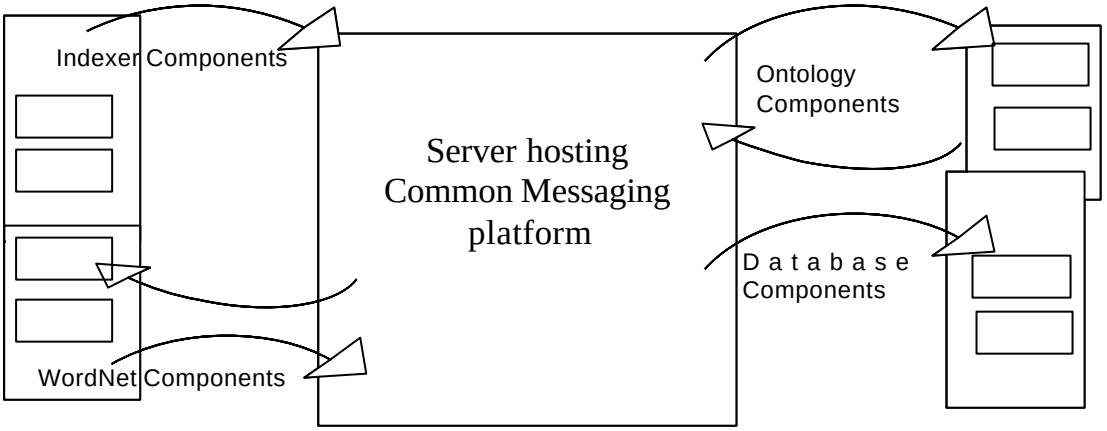

Figure 3: Common Messaging Platform

There can be one or more instances of various components such as Indexers, WordNet servers, UTON components and Database components. A component such as an indexer may in turn have more than one instance. For example, a video indexer may be running three of its instances and an image indexer will have two instances.

We have implemented a communication protocol in architecting the common messaging platform that no component communicates directly with any other component. All communication between common messaging server and client components are in the form of messages. For example, the video indexer sends a message to common messaging server requesting synsets for each of the terms it obtained. The common messaging server sends a message to the available instance of WordNet component giving the particular term as part of the message and the WordNet component gives back the synset back to the common messaging server and so on.

Multimedia indexers would parse the corresponding media objects from the documents and then return a set of terms. This is the key point because the entire complexity of the media object analysis process is inside the indexer and whatever may be the media object, the indexer returns only the terms (basically some textual strings) which will be easier to pass between the components. If we had to pass the actual multimedia objects it would result in an unusable system because we will not be able to achieve reasonable response and processing times. If actual objects need to be shared by different components then only the location addresses will be passed and those component will directly interact with the storage tracking system. In this manner, we achieve high performance and high scalability.

4. Conclusions

The importance of building highly scalable and feature rich content management and information management systems is becoming very clear as we face the challenges of information overload. We have discussed common features of current generation content management systems and the trends of new generation systems that demand rich media processing, conversation based interfaces and personalization aspects. We have tried to identify the key functionality of future content management system and the technology enablers to achieve this functionality. Language processing and rich media analysis will play a major role in making future content management systems more effective but, they are far from being mature disciplines. With the existing technology know-how, we have tried to present two enablers that form a part of the foundation for building next generation content management systems.

One technology enabler is to build a Uniform Taxonomy and Ontology Network (UTON) where multiple taxonomy or ontology belonging to different domains can co-exist with uniform interfaces. This will help us in building enterprise wide content management systems that can be used across departments, locations and service or product offerings. UTON can aid in creating metadata of the content, content navigation and in scaling the content and information management systems. Second technology enabler is creating common messaging platform using component architecture so that very complex and high scale content management systems can be built with high levels of performance and with minimum communication overheads. The first technology enabler will help in improving the functional aspects of content management system as it addresses quality of content characterization and the second enabler will help non-functional aspects such as scalability and performance.

5. References

1. Demoz, http://demoz.org
2. Fellbaum, Christiane (Ed). WordNet: An electronic lexical database, MIT Press, 1999.
3. Feldman, Susan “Content Management” in eInform Volume 3, Issue 7, IDC News letter, 2001
4. Lenat, D. B. and R. V. Guha. *Building Large Knowledge Based Systems*. Reading, Massachusetts: Addison Wesley, 1990.
5. Price Waterhouse Coopers, Technology forecast 2003-2005. 2003
6. Harabagiu Sanda M, Moldovan Dan I, Knowledge processing on an extended WordNet appeared in [Fellbaum, 1999]
7. Semantic web: http://www.semanticweb.org

About Author

Dr. Vasudeva Varma is a faculty member at International Institute of Information Technology, Hyderabad Since 2002. Prior to joining IIIT-H, he was the president of MediaCognition India Pvt. Ltd and Chief Architect at MediaCognition Inc. (Cupertino, CA). Earlier he was the director of Engineering and research at InfoDream Corporation, Santa Clara, CA. He also worked for Citicorp and Muze Inc. in New York as senior consultant. He obtained his Ph.D. from the Department of Computer and Information Sciences, University of Hyderabad in 1996. He has five patent applications and several publications in journals and conferences. He recently obtained young scientist award and grant from Department of Science and Technology, Government of India, for his proposal on personalized search engines. His areas of interests include search and information extraction, knowledge management and software engineering. He is heading the Search and Information Extraction Lab at Language Technologies Research Center (LTRC). His team is developing search engines for Indian languages, working on named entity extraction and personalized search engines. He is also interested in experimenting with non-conventional methods for teaching software engineering in general and case study based approach In particular.

Email : vv@iiit.net