

The Application of Semantic Enablers in the Context of Content Management Systems

Samuel Sambasivam
Azusa Pacific University,
Azusa, CA, USA

ssambasivam@apu.edu

Tilemachos Kanakis
First Consulting Group (FCG)
Manhattan, NY, USA

tkanakis@fcg.com

Abstract

Content management can be viewed as the concepts, processes, technologies and practices that deal with the development, management and publishing of information. Traditionally, a Content Management System (CMS) does not offer the means for content classification, management and retrieval of data based on dynamically defined information models. Furthermore, Content Management Systems usually rely on indexed text, syntactic and/or structural metadata for content classification and retrieval. The paper discusses the design of an experimental Semantic CMS (SCMS), which exercises the application of semantic enablers – ontologies, classification & metadata – in order to achieve capture and utilization of content meaning. Lastly, it suggests that the application of semantic enablers in the context of Content Management Systems can provide the means for better information organization and retrieval as well as enhanced machine interoperability.

Keywords: Content Management, Semantic enablers, Classification, Ontology, Metadata, OWL, Relational Database

Introduction

Content management can be viewed as the concepts, processes, technologies and practices that deal with the development, management and publishing of information. However, traditional Content Management Systems rely on indexed text and syntactic or/and structural metadata for information classification and retrieval. It is argued (Fisher & Sheth, 2003) that these techniques do not promote the use of contextual meaning of stored information and they point out three basic challenges for Content Management Systems. Firstly, the heterogeneous data sources that many large-scale information systems have to comprise, where the stored data differ structurally and syntactically. This reveals a need for normalizing the representation of data in order to enable their interoperability and also make it equally accessible by humans and machines. Secondly, data is not only stored in heterogeneous data sources but many times in distributed data sources among

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

various machines on possible different networks. Any system that aims to share content will need to take into consideration technologies that can securely provide generic transport methods and allow information integration, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP) and Web Services. Lastly, it is the relevance factor. Probably, the most de-

manding challenge is to provide the means to the users of a CMS to drill down to the most relevant for the task information, which can exist in large data storages. Ideally, a CMS should provide the same or increased quality of data management as the data grows. However, it appears that this is not often the case. As the data grows the quality of retrieved information usually degrades due to the inability of the system to retain and use contextual meaning about its content and/or explicitly define and interpret the intentions of the user. As a result, information retrieval is inefficient and users are expected to act upon irrelevant results in order to locate the information of interest. It could be argued that the inability of traditional Content Management Systems to organize content in a manner where content can carry meaning and not only data is the major missing element from the current CMS model. This paper presents the design of an experimental software solution to the problem of content management via the application of semantic enablers. In particular, the solution utilizes metadata and ontology based classification technique to achieve better information organization and retrieval.

Motivation

Ideally, a CMS should provide appropriate mechanisms to organize and publish content in structures that can carry meaning and not simply data. Ironically, the World Wide Web (WWW), which could be considered as the largest and continuously growing source of data that is not structural consistent, has faced the same challenge. One of the proposed solutions for this problem is to represent Web content in a form where content can carry machine-understandable meaning and use intelligent techniques to take advantage of these content representations. This revolutionizing plan is known as the Semantic Web initiative (Berners-Lee, Hendler, & Lassila, 2001). The proponents of Semantic Web have proposed Web Ontology Language (OWL) that is an XML based technology capable to describe contextual information and capture referential knowledge representation. This in turn can allow machine-based agents to utilize this representation, operate upon the Web content and retrieve information within particular domains of knowledge, transforming in this way the Web in a more powerful and intelligent resource.

Furthermore, classification techniques are usually employed to increase the relevancy factor of information. Several classification techniques for textual and non-textual content are based on statistical analysis, pattern recognition, machine learning and rule-based methods to address content classification (Breen, Khan, Kumar & Wang, 2002; Frasconi, Soda & Vullo, 2002; Ipeiritos, Gravano & Sahami, 2000; Joachims, 1998; Yang, 1998). Moreover, ontology-based classification techniques can be used for information classification too. Studies have shown how classification results can be more precise when classified documents have XML based metadata that conform to predefined schemas (Fisher & Sheth, 2003; Sheth et al, 2003). The role of XML based semantic metadata could be considered invaluable. It could be viewed as the building block of semantic content and offer the means for better classification of information, high-precision information retrieval and increased interoperability among heterogeneous systems.

Content Management systems could clearly benefit from semantic enablers such as ontology based classification and XML based metadata. The fundamental idea behind ontology oriented modeling of an information domain is the ability to explicitly and efficiently represent classes of a domain and important associations among these classes. These associations can be utilized to traverse or search a content storage according to predefined content classes whose structure of properties can be much deeper than the basic taxonomy. The “labeling” of relationships offered in ontology based classification gives the opportunity to associate classes of content in meaningful ways and provide the necessary for information retrieval ambiguity resolution. As a result, information about the context and relationships of a content piece can be preserved and increase the relevance factor among different content pieces enabling more efficient information retrieval.

Synopsis of the SCMS

It was envisioned that a SCMS should enable ontology engineering as the means to model an information domain, and also support creation, storage and retrieval of content according to the underlying ontology concepts. In addition, it should facilitate machine interoperability. This should be achieved by exposing both the information model and stored content in OWL format, preferably via a Web Service, where other applications can utilize to retrieve and process information. Lastly, demonstration of reasoning mechanisms, which could allow process of content in meaningful for the users' manner, should also be supported, but this feature fell outside the scope of this experimental development.

In particular, the system was composed of six sub systems providing with the following assigned responsibilities:

- *Presentation Subsystem*
Encapsulates graphical user interface and presentation services that allow to a) carry out an Ontology engineering process in order to model an information domain, b) a content form editor to adjust the presentation aspect of available content classes, c) create, modify, browse and search content according to the underlying ontology concepts and their associations.
- *Content Model Management Subsystem*
Concepts in an information domain can be described utilizing semantic axioms adopted from OWL DL. This subsystem encapsulates business rules that enforce these semantic formalisms and maintain model integrity. It is also responsible for the synchronization of content metadata with actual content storage structure.
- *Content Modification Subsystem*
It provides the necessary functionality for the creation and management of content based on a predefined information model. It enforces business rules necessary to maintain content versioning and integrity.
- *Content Delivery Subsystem*
It provides access to content in order to browse and search for stored information. A web service wraps this functionality where content and metadata are delivered in OWL DL format.
- *Security Management Subsystem*
It provides mechanisms for user authentication and authorization. It also manages user accounts, roles and role permissions on different classes of content.
- *Data Management Subsystem*
It provides data access mechanisms in order to store and retrieve metadata and content. It is also responsible for the generation of appropriate SQL code utilized to dynamically create the underlying relational database schema for the information domain being model.

In addition, this experimental development was carried out using C#, MS.Net Framework and MS SQL Server 2000 as the backend database.

Adopting OWL Formalisms

OWL DL formalisms (Bechhofer, Harmelen, Hendler, Horrocks, McGuinness, & Patel-Schneider, 2004; McGuinness & Harmelen, 2004; Smith, Welty, & McGuinness, 2004) were adopted for the design of the system. The main purpose of this approach was to enable modeling of an information domain utilizing semantic formalisms that can be easily transformed and exposed for ma-

chine processing. In brief, the system enables the user to create taxonomies of content classes. A content class can be one of the following types:

- *General content class*, which is similar to an OWL class.
- *Abstract class*, serve as the base of more specific classes of information and without any direct instances.
- *Enumeration class*, are used to define a class of information with a finite set of class instances, which is similar to an OWL Enumeration class.
- *Boolean combination of classes* that are Union class, Intersection class, and Complement class. These types do not have any direct instances but the concept deviates from OWL formalisms.

A Boolean class is always a subclass of another general content class. Their purpose is to create meaningful Boolean combination classes that can efficiently group together instances of their base content class. Boolean combination classes represent a more advanced class constructor in Description Logic, which is a representation of AND, OR and NOT logical operators on classes. The definition of a union class includes a list of other classes, the union members. The union members are used to define a class of content where its instances will be all those instances that appear at least once in the union members. The members of an intersection class define a new content class where its instances will be all those direct instances that are common to all intersection members. Finally, a complement class defines a content class where its instances are all the direct and indirect instances of its base class except those of its complement. An abstract class cannot not be part of Union, Intersection or Complement class definition since the purpose of these classes is to group together (or exclude) direct instances from other content classes. Furthermore, properties can be added to taxonomies and used to state relationships between class instances and data values. Adopting again OWL DL formalisms two types of properties were identified:

- *Data type property* that defines a relationship between an instance of a class and an XML Schema data type.
- *Object property* that defines a relationship between two class instances.

The system allows the user to view the property's *domain* and define property's *range*, *characteristics*, its *base* property and synonyms used for the property. *Domain* is used to limit the classes where the property is applied. This feature can be viewed as a global restriction because the domain of a property is stated on the property and not on the class that is associated with it. Properties are applied to classes by defining a property facet for the class. In addition, property facets encapsulate cardinality restrictions and other facet information. Considering that an object property aims to define a relationship between class instances, there was a need to define ownership of this relationship in order to facilitate appropriate data integrity. In other words, it must be explicitly defined whether the object property facet determines a relationship owned by the class where the property is applied or it simply links two class instances. Using this ownership construct the system enforces creation of a new class instance when a new value is created for the applied *object property*. When the value is removed then the system cascades deletion of instances the removed value owns. The system also deletes any links other instances may have to the instance being deleted. This ownership construct, not apparent in OWL DL, has been considered essential in order to maintain content integrity. *Range* defines the type of property's value. If it is a data type property then it will link the property to a single XML Schema data type. In a hierarchy of data type properties a child property and its base have the same *range*. If it is an object property then the range links the property to a content class that defines the property's value type. Object properties inherit and extend the range of their base.

Property characteristics provide a powerful mechanism in order to enhance reasoning about properties and their values. They can be applied in order to provide more meaningful semantics, which could be used by other applications to carry out inferences about stored content. The system allows defining property characteristics according to OWL DL formalisms. In OWL Full data-type properties are a subclass of object properties. However, in OWL DL data-type and object-type properties are disjoint by definition. This implies that property characteristics *inverse-of*, *inverse-functional*, *symmetric* and *transitive* can be only specified for object properties. Moreover, both data type and object properties can be characterized as functional.

Meta-Model Principal Design Classes

The class model in Figure 1 illustrates principal system classes that aim to describe metadata information. Although the class model has adopted some semantic formalisms from OWL DL, it is not a precise illustration of these formalisms.

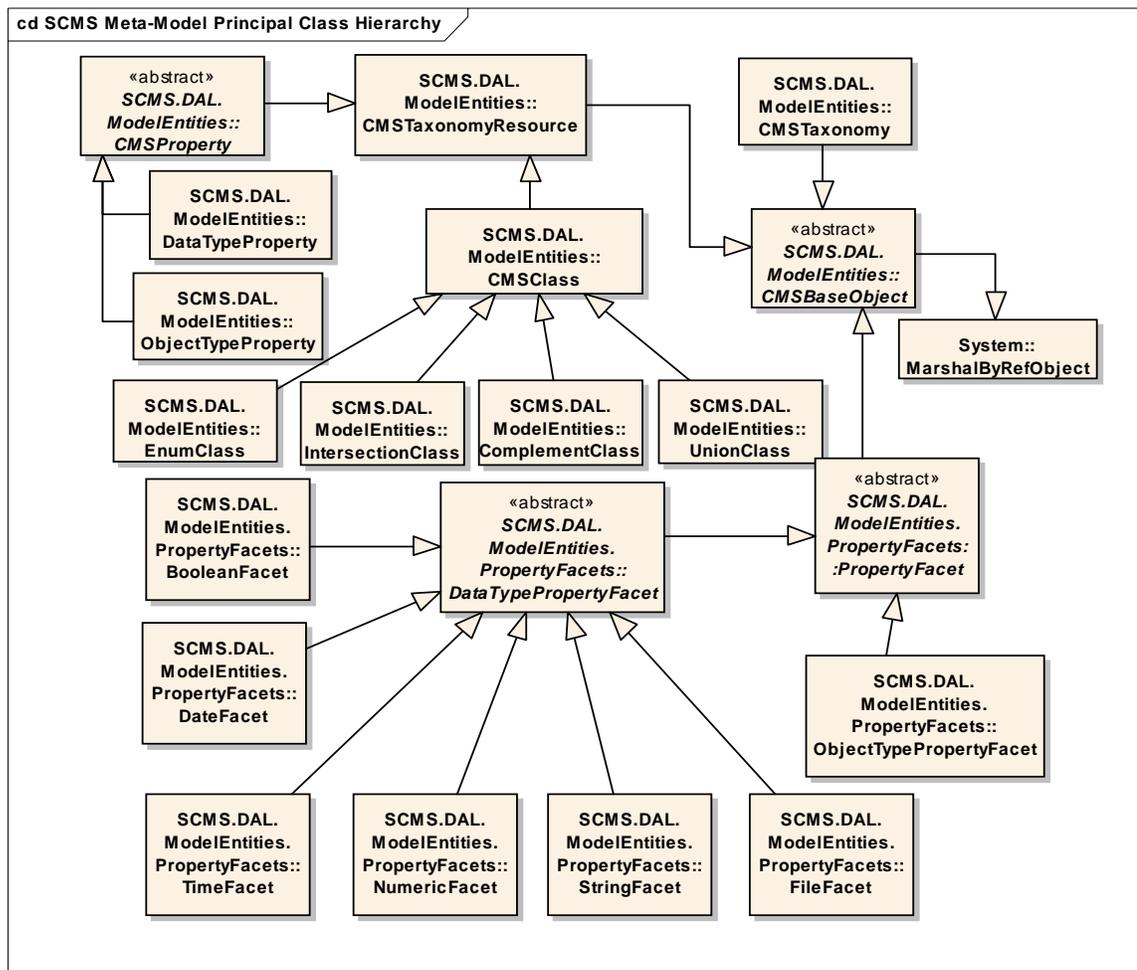


Figure 1 – Meta-Model Principal Class Hierarchy

Figure 2 represents relationships between the principal system's classes. These associations reveal several differences between OWL DL formalisms and what has been applied to cover the needs of this project. Firstly, the system is designed to support only single inheritance for content classes and properties. Moreover, *CMSClass*, *DataTypeProperty* and *ObjectTypeProperty* classes explicitly define their subclasses and sub-properties respectively, which is not the case in OWL DL. Also, the class model does not capture the concept of equivalent classes and properties, but

out the class’s life. The instance version table contains attributes that reflect data type properties applied to the class and therefore it can change accordingly. The code also implements generation of a table for each class’s multi-value data type property. A single table is also generated for each object property in order to store relationships between class instances. In particular, the structure of an object property table facilitates the need for an instance version to always reference the active version of another instance. Database views are generated for each Boolean combination class. Figure 3 illustrates the mapping of taxonomy’s classes and properties to a generalized database model.

Another interesting aspect of the implementation was the development of the system’s search capabilities. The underlying database schema is not static and therefore queries are composed on the fly using content metadata and user defined filters on data type properties. The system decides what database tables need to be queried depending on the classes and data type properties the user has selected. Finally, the query is translated to an appropriate SQL statement in order to retrieve results from the database.

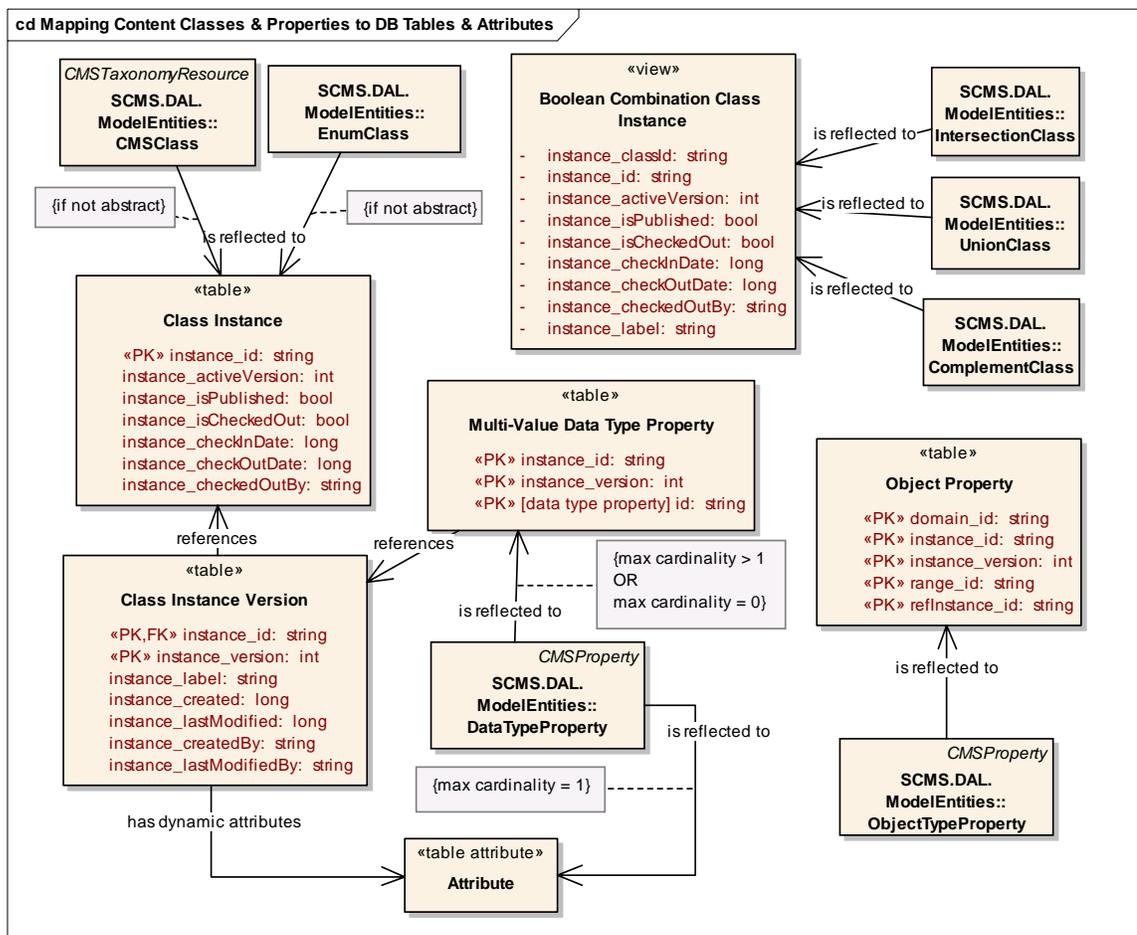


Figure 3 – Mapping Content Classes & Properties to DB Tables & Attributes

Different types of searches generate different query structures, meaning that the structure of an SQL statement for a search through the class hierarchy will be different from the one generated to search content via object property hierarchy. A user can traverse and query content using either a class hierarchy or data type property hierarchy or an object property hierarchy. These search capabilities could be summarized as it follows:

- Search of content utilizing a class hierarchy: Filter data type properties applied on a selected class. If the class is defined as abstract then the results include all instances from its subclasses.
- Search of content utilizing a data type property's domain: The user can submit a query for a single data type property and select to retrieve results only from specific content classes where the property has been applied.
- Query an Object Property: The user can query content by filtering data type properties applied on the *range* of an object type property and retrieve those class instances that reference the results of the query. For example, let's assume a scenario where we have defined a relationship between classes *Course* and *Professor* using the object property *TeachesCourses* (where *Course* is the range and *Professor* the domain). The user can filter data type properties applied on *Course* class in order to retrieve those instances of *Professor* class that reference one or more instances of *Course* class in the query's results.

Furthermore, it should be highlighted at this point that the execution cost of a query might vary depending on the type of the class and the number of classes used to retrieve the results. For example, querying a class means that the results can be a union of queries on that particular class and each child class. This in turn could imply a linear relationship between the execution cost of a query and the depth of class hierarchy (or number of classes) being queried.

Trends & Related Work

The primary driving forces behind semantic technologies, which aim to make the vision of Semantic Web a reality, is to juxtapose web enabled content and content semantics in order to bring information processing closer to human thinking. Researchers and organizations have realized the importance of semantics in several ways. They have been involved, among other things, in the development of domain specific ontologies and metadata standards, organization of content according to such taxonomies and annotation of content with semantic metadata. Antoniou and Harmelen (2004) point out the contribution of semantic Web technologies in businesses such as Audi and Swiss Life for content organization and retrieval as well as data integration. The concept of ontology as the context for semantics and its use to access content has been discussed before. Semantic Content Organization and Retrieval Engine (SCORE) (Sheth, Bertram, Avant, Hammond, Kochut & Warke, 2003) technology supports automatic semantic metadata extraction out of content and ontology-based content classification. Another important aspect of SCORE is that provides information retrieval in XML format, however, not in standard OWL format. OntoWeb (Spyns, Oberle, Volz, Zheng, Jarrar & Sure, 2002) that is a semantic portal where knowledge can be gathered and retrieved using ontology based annotated information. OntoWeb offers content provision, which is achieved using automated content syndication or manual content creation by portal users. The portal also offers content browsing and search according to the underlying concepts in OntoWeb ontology. Furthermore, Haystack (Huynh, Karger, & Quan, 2002) is a Resource Description Framework (RDF) based information management system. It provides users with a flexible way to describe, organize and retrieve information as well as to delegate processing tasks to software agents. The system provides an ontology to support collection-based organization, semantic categorization, and collaboration and trust management. Gauch, Chaffee & Pretschner (2003) report on research on technology that adapts to user web browsing habits. Users can create their own concept hierarchy in order to access content, or the system can create an ontology by observing the user's navigation habits. Lastly, it has been claimed (Bussler, Fensel, & Maedche, 2002) that semantic enabled Web Services can transform the Web from a static collection of information into a machine process-able and machine interpretable distributed device.

They propose a design for such a system, which is based on a Web Service Modeling Framework that provides a model for describing and developing semantic Web Services. Ontologies are in the heart of this model used to describe linked concepts and to achieve the required machine interoperability.

Conclusion & Future Directions

In conclusion, it could be claimed that ontology based techniques can be utilized for classification of content in order to increase the relevance factor of information. Content Management Systems can implement such techniques to gain an advantage over their traditional counterparts that simply rely on indexed text and syntactic and/or structural metadata for information classification and retrieval. A CMS can support ontology engineering to model an information domain and use model's metadata to generate and query the content storage. OWL can be utilized to publish model and content in a standard XML format. OWL could be considered as the means to represent data, data structures and important relationships among them in a machine process-able format in order to enhance systems interoperability.

It could be claimed that from a knowledge-acquisition perspective it would be very convenient to exploit semantic information, such as class and property hierarchies as well as property characteristics. An inverse, symmetric or/and transitive relationship could definitely be utilized to infer property values that have not been explicitly applied to a class instance. Further work is required in order to determine how such functionality could be added. Furthermore, it could be claimed that the application of semantic enablers gives an obvious advantage for information retrieval. In this development, a semantic layer was created and used to generate appropriate database tables for content storage. It could be assumed that the inverse should be feasible too. Building a semantic layer on top of one or more existing databases in order to provide a normalized view of content from different database tables should be also a realistic scenario. This could be viewed, to a certain extent, as the reverse process of what the current system carries out. That hypothetical process could involve gathering tables and their attributes from a database schema. Tables could be mapped to content classes and their attributes to appropriate data type properties. Object properties could reflect relationships between tables. More general content classes and properties can be defined and utilized to create class and property hierarchies, where classes and properties directly derived from the database schema are the leaves in these hierarchies. A content portal could use such semantic layers to provide access to content from different databases and enable users to drill down to the most relevant information for a specific task.

References

- Antoniou, G., & Harmelen, F. (2004). *A semantic Web primer*. Cambridge: MIT Press.
- Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. et al, (2004). OWL Web ontology language reference. Retrieved February 10, 2005, from <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
- Berners-Lee, T, Hendler, H. & Lassila, O. (2001). The semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Retrieved February 15, 2005, from <http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>
- Breen, C., Khan, L., Kumar, A. & Wang, L. (2002). Ontology-based image classification using neural networks. Retrieved January 29, 2006, from http://escsb2.utdallas.edu/ORES/papers/Spie_Paper2.pdf
- Bussler, C., Fensel, D. & Maedche, A. (2002). A conceptual architecture for semantic enabled web services. Retrieved January 29, 2006, from <http://www.sice.umkc.edu/~leeyu/class/CS690L/Reference/4.Bussler1.pdf>

- Fisher, M. & Sheth, A. (2003). Semantic enterprise content management. Retrieved October 21, 2004, from <http://www.semagix.com/documents/SemanticEnterpriseContentManagement.pdf>
- Frasconi, P., Soda, G. & Vullo, A. (2002). Hidden Markov models for text categorization in multi-page documents. Journal of Intelligent Information Systems. Retrieved January 29, 2006, from <http://www.dsi.unifi.it/~paolo/ps/jiis-02-multipage.pdf>
- Friedlein, A. (2003). *Maintaining & evolving successful commercial Web sites: Managing change, content, customer relationships, and site measurement*. San Francisco: Morgan Kaufmann.
- Gauch, S., Chaffee, J. & Pretschner, A. (2003). Ontology-based personalized search and browsing, Web intelligence and agent system. Retrieved January 29, 2006, from <http://www.itc.ku.edu/~sgauch/selectedpapers/WIAS2003.pdf>
- Huynh, D., Karger, D. & Quan, D. (2002). Haystack: A platform for creating, organizing and visualizing information using RDF. Retrieved December 5, 2004, from <http://semanticweb2002.aifb.uni-karlsruhe.de/proceedings/Research/huynh.pdf>
- Ipeirotis, P., G., Gravano, L. & Sahami, M. (2000). Automatic classification of text databases through query probing. Retrieved January 30, 2006, from <http://persival.cs.columbia.edu/papers/webdb2000.pdf>
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Retrieved January 29, 2006, from http://www.cs.cornell.edu/People/tj/publications/joachims_98a.ps.gz
- McGuinness, D. L. & Harmelen, F. (2004). OWL Web ontology language overview. Retrieved January 16, 2005, from <http://www.w3.org/TR/owl-features/>
- Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K. & Warke, Y. (2003). Semantic content management for enterprise and the Web. University of Georgia. Retrieved January 20, 2005, from http://lsdis.cs.uga.edu/SemWebCourse_files/SCORE-draft.pdf
- Smith, M. K., Welty, C. & McGuinness, D. L. (2004). OWL Web ontology language guide. Retrieved January 15, 2005, from <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>
- Spyns, P., Oberle, D., Volz, R., Zheng, J., Jarrar, M., & Sure, Y. et al (2002). OntoWeb – A semantic web community portal. Retrieved January 16, 2005, from http://www.starlab.vub.ac.be/staff/mustafa/publications/2002_pakm_ontoweb.pdf
- Yang, Y. (1998). An evaluation of statistical approaches to text categorization. Retrieved January 28, 2006, from <http://reports-archive.adm.cs.cmu.edu/anon/usr/ftp/usr0/ftp/1997/CMU-CS-97-127.ps>

Biographies



Dr. Samuel Sambasivam is the chairman of the Department of Computer Science of Azusa Pacific University. Professor Sambasivam has done extensive research, publications, and presentations in both computer science and mathematics. His research interests include optimization methods, expert systems, Fuzzy Logic, client/server, Databases, and genetic algorithms. He has taught computer science and mathematics courses for over 20 years. Professor Sambasivam has run the regional Association for Computing Machinery (ACM) Programming Contest for six years. He has developed and introduced several new courses for computer science majors. Professor Sambasivam teaches Database Management Systems, Information Structures and Algorithm Design, Microcomputer Programming with C++, Discrete Structures, Client/Server Applications, Advanced Database Applications, Applied Artificial Intelligence, JAVA and others courses. Professor Sambasivam coordinates the Client/Server Technology emphasis for the Department of Computer Science at Azusa Pacific University.



Tilemachos Kanakis is an IT professional and currently an associate with First Consulting Group (FCG) in Manhattan, NY, USA. He has seven years of experience in analysis, design and development of desktop standalone and distributed applications utilizing Microsoft Technologies. His professional background is focused on Internet technologies and development of content management, workflow and reporting systems in banking and health sector. He has practical exposure to several development methodologies, processes and tools, such as RUP, XP, Iconix Process, UML, C++, C#, VB, ASP, MS .NET Framework, MS SQL Server, DB2, etc. He has a great interest in Semantic Web and emerging technologies as well as applications of AI and traditional statistical techniques for data analysis. During his academic pursuit for M.Sc. at Liverpool University he had the opportunity to further study and explore a variety of interrelated IT areas such as AI, Internet development, metadata technologies and current trends in software engineering.