An Ontology-based Approach to Machine Learning and Distributed Knowledge Management

Kevin Deeb, Ph.D.

Associate Professor and Academic Coordinator of Information Technology
Barry University, Miami Shores, FL 33161
kdeeb@mail.barry.edu

ABSTRACT. This paper proposes a novel approach to knowledge discovery and network adaptation through a high-level ontological and context-based architecture that facilitates information customization and knowledge organization through a longitudinal study of user/network behaviors. The proposed model aims at managing distributed knowledge items through stand-alone computational layers that use ontology to describe and represent knowledge as well as context to adapt knowledge to its hosting environment.

KEYWORDS: Ontology, Machine Learning, Knowledge Management (KM).

(Received January 19, 2006 / Accepted August 03, 2006)

1. THE KNOWLEDGE WAVE

Semantic web aims to organize disparate data and improve information usability by building an information layer on top of data architecture. The semantic web concept is manifested in a system of standards organized into three operational layers. The base layer, defined as W3C's RDF (Resource Description Framework), uses a metadata language for representing information architecture. RDF uses meaning description tags to enable the reusability of information across the Web. Using the tags, the data on a web page is wrapped with meaning and converted into information. The meaning of the tags is described in an application specific class-based schema, located in a separate file, and referenced in the web page. The information schema, hosted at the second layer, is described using a separate schema description language called RDFS. The actual definition of the meaning and semantic co-relations are managed on the third layer through ontologies using W3C's OWL (Web Ontology Language). Ontology is used in this research to describe knowledge representation and structure.

Nevertheless, knowledge is only meaningful within the context of its environment. For instance, [1] gave an epistemological view of knowledge where he noted that "this perspective defines knowledge as the social practice of knowing". It suggests knowledge to be embedded in a community, supersede any one individual, and to be highly context dependent. With such a perspective on knowledge, the sender-receiver model is no longer an appropriate conceptualization of the knowledge sharing process. The other supporting epistemological view of knowledge is "that which is known" and embedded in the individual (i.e., contextualized knowledge). This view refers to the indivisibility of knowledge from its ontology on the personal level and perceives knowledge sharing as "the exchange of information in order to yield knowledge" [1].

In Knowledge Management systems, ontology has been used to capture implicit knowledge of the knowledge workers and to associate it with knowledge artifacts for classification, search, and browsing purposes. Both, the knowledge artifacts and the ontology, are managed in a P2P style that resides on a node in the P2P system and acts collectively as client and server. Each node can directly communicate with other nodes (i.e. building a P2P network) for acquiring and retrieving knowledge [4].

While this paper does not address the semantic web architecture, it proposes a communication model that extends the semantic web with knowledge and communication layers. These layers host and manage ontologies and contextual information. Ontologies have been used in this research to represent concepts and knowledge structure, whereas context reflects knowledge adaptability to its hosting environment. Context augments information with contextual references, therefore personalizing information search and retrieval.

2. FRAMEWORK SPECIFICATIONS

Ontologies are used in this research to support the integration and management of knowledge. To seek interoperability and compatibility among information scattered in the global network, ontologies play a major role in bringing together disparate information and knowledge items. However, to be successful, it should incorporate human-like intelligence functions, knowledge mapping, and social communication constructs in a single integrated setting. This holistic framework must provide a store of manageable knowledge and network resources on which to share existing and build new knowledge. It must also recognize the individual and social dimensions (e.g., needs and expertise identification), and provide processes, tools, protocols, and interfaces to foster knowledge sharing and acquisition. In effect, the framework should be composed of four stand-alone phases:

- **The Initiation Phase**: Involves capturing individual and social cognitive specifications (e.g., needs analysis), and system & network resources (i.e., environment) using context constructs (aka internalization constructs).
- **The Validation Phase**: Focuses on information retrieval & acquisition and knowledge validation through the use of both information and ontology constructs.
- **Construction & Integration Phase**: Produces and integrates all information and knowledge items, semantic relationships, and constraints into an ontology-based schema through the use of ontology constructs.
- **The Communication Phase**: Includes the deployment and communication of valued knowledge to shared parties through the use of communication constructs.

Ontologies are stored at a knowledge node (KN) and locally produced and manipulated by the KN owners (KN host). Each KN facilitates information exchange and knowledge management across different KNs without assuming shared meanings. This would enable the translation of different meanings using protocols and interface languages managed locally at each knowledge layer. Also, the autonomy of the KN enables semantic classification and identification of information and concepts. A KN node may interact with other nodes in the network for knowledge sharing and evolution.

The KN represents ontologies as a network of entities (concepts and resources). An entity is characterized by a set of attributes and a set of relationships with other entities. The entity may be tagged with an absolute or relative location (location attribute). Ontology of a KN is dynamically updated by adding new entities or enriching it with more descriptions (e.g., attributes, relationships, user preferences, annotation, or relevance). Relationships between entities are dynamic indexes that contain not only a linkage to other entities in the network, but also relationship attributes to denote KN-specific information (e.g., ranking). The process of updating ontologies may be automatically performed using OWL. Due to the autonomous and self-organizing nature of the KN system (e.g., capturing, sharing, and leveraging knowledge), it has the potential to stimulate the acquisition and presentation of knowledge.

Through the support of ontologies, this framework can offer learners, educators, practitioners, professionals, and researchers tools to support information exchange and knowledge accessibility in several ways, including:

- Ready access to organized information and subject experts
- Better communication and interaction tools that support collaborative solutions
- Access to personal knowledge support and search tools
- Use of specific solutions to identified problem domains
- Access to materials from various sources and group-based solutions to problem domains
- Information retrieval and access permissions and security
- Information evaluation, classification and customization using knowledge ontologies
- Information inference and conversion to knowledge
- Collaborating around information
- Knowledge acquisition and exchange

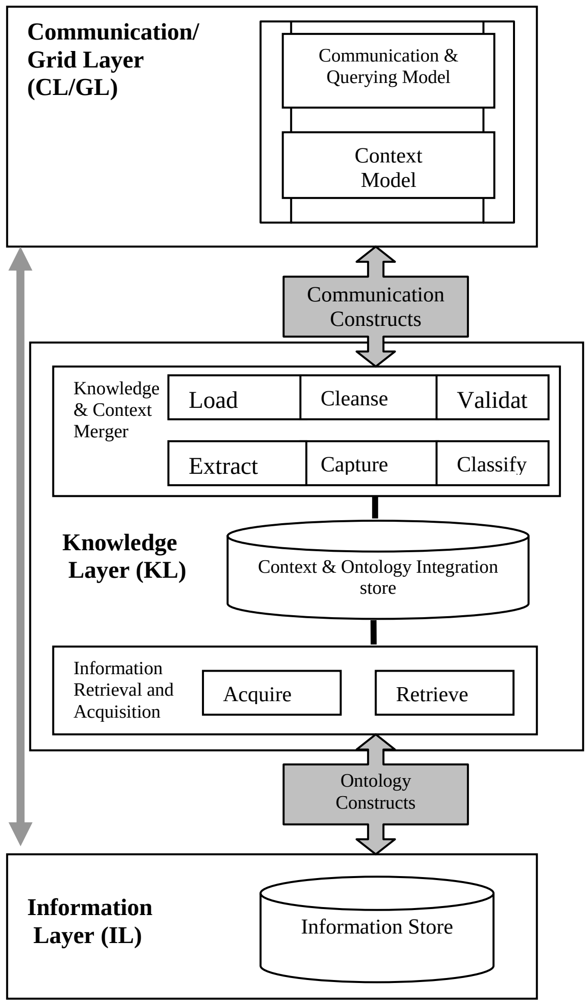

Figure 1: Overview of KN Architecture

The following sections further discuss the design of the KN framework to support the abovementioned requirements and specifications.

3. SYSTEM CONCEPTUAL ARCHITECTURE

The KN, as shown in Figure 1, is composed of three main layers. Each layer may pose a request to its neighboring layer through a function with parameters that are native to its own domain, and gets results back in its native language. The layered approach presents a well-defined functional architecture to acquire and process information and knowledge by projecting relevant information into the corresponding ontology. Each KN publicizes its own knowledge using OWL or RDFS. The following subsections describe the components of the overall framework.

3.1 Communication/Grid Layer: Context, Presentation, and an Integrated view of Network Resources.

The Communication layer (CL) is structured into two models: the Context and the Communication & Querying (C&Q) Models. The Context model is responsible for content mapping and message generation, whereas the C&Q model is mainly concerned about how the chosen content should be presented. The Context model continually stores information about user activities throughout a live session. The C&Q model solicits and interfaces its services and knowledge domain(s).

The CL collects and maintains information regarding user interactions, preferences, and configuration through the support of data mining. It hosts the query model (searching and retrieving) and defines the rules and protocols for nodes interaction and communication, negotiation, processing, and security (i.e., RDFS). This layer is also equipped with a query modeling tool to submit and process queries by comparing it against the underlying ontologies. When queries are received, they are processed and disserted according to the corresponding ontology schemas.

The CL interacts with the underlying *Knowledge Validation* component to cleanse and parse queries prior to passing the results. This layer acts as a font-end interface facilitating information flow and supporting both interaction and intersection. This layer has also the following characteristics:

- Group dynamics: Facilitates interaction and collaboration among participants using an outlined scenario or method.
- Classifications: Encourages customization of information based on user preferences and specifications, defined in the *User Ontology*.
- Change behavior: Observes the learners' reactions as a result of the interaction and adapts its behavior accordingly, defined in the *Context Ontology*.
- Design environment: Supports collaborative interaction with the task(s) and/or the interaction between the learners defined in the *Grid Ontology*

.

The top layer acts a grid-based system and thus named the Grid Layer (GL). Grid computing is defined as “a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations” [3]. Grids enable heterogeneous systems to interoperate in sharing and combining resources resulting in new resources that are tailored transparently to a specific user domain group. Grids are unique to ontology creation because of its access to all the resources and its ability to assemble the necessary resources and create a virtual network [3, 4].

The knowledge enabling power of grids comes from its own ability to create virtual environments, known as virtual enterprises (VE). The VE is defined in this research by available resources & features, permissions, views, users, and any other attributes custom to the topic (problem) that requires the VE. Information from all over the grid is then projected into the virtual ontology space, Grid Ontology, only in the dimensions that are needed to solve the particular problem of interest. This makes a grid a far more advanced information technology system than any distributed computing system, which simply enables resource sharing across a defined domain (i.e., context settings). It focuses on combining the resources for all the users of the VE. The GL creates new resources, composed transparently of all the combined resources, accessible to the users via a standard interface, through the Communication Model, that is independent from the interfaces of each of the member resources.

The GL builds a VE based on the posed request, and then project into this VE only information and resources that are related to the request (very similar to views in DB models). This information in the VE then becomes problem applied. Therefore, the projected information becomes knowledge for the request. The VE, serving as a Grid Ontology, models and hosts the resources based on the request/problem at hand.

Thus, the GL creates on-demand multifaceted knowledge structures that pave the way for generation and management of recourses that are coordinated within the underlying Knowledge layer (KL). For a specific problem at hand, the Grid projects selected member resources into a virtual environment specific to the problem (using the Grid Ontology). Out of the information that will be fed into this ontology, it has innate capability to model and host knowledge that is specific to the problem at hand. Through this process, the Grid Ontology becomes dynamic that interacts with the underlying layer and neighboring nodes to build and maintain an integrated view of subsets of KNs in the network that fulfill the domain specific requirements.

3.2 Knowledge Layer: Context and Knowledge extension of the underlying Information Layer

Creating ontologies from contextual information is the main functionality of the Knowledge layer (KL). The KL acts as a coordinator to parse user query information. Once the semantics of the information is captured and transferred to knowledge, the KL manages the same knowledge and maps semantic information and metadata to resources and user-defined ontologies (e.g., User and Grid Ontologies).

To enrich information with knowledge, this layer maintains ontologies describing the knowledge of the node. Each entity of an ontology may correspond to one or more entities that are linked using relationships managed by the underlying Information Layer (IL). For example, the entity Hotel may correspond to the Location entity that is related to the Accommodation entity. An entity may be rated by the user (i.e., preferences, discussions, and expert rating).

The items of an ontology could be selected based on the request domain, using knowledge bases, directory servers, associations, etc. Ontology items include inference rules, classes of meaning, object relationships, inheritance rules, set rules, etc. The ontology creation process resides fully in the KL. The ontology it creates, however, becomes a connection between each the Information and the Knowledge layers.

The KL represents a global ontology, which is an integrated view of specialized ontologies that are discovered in collaboration with the GL. Global ontologies are domain-specific ontologies that represent personalized, collected yet filtered knowledge about specific domains.

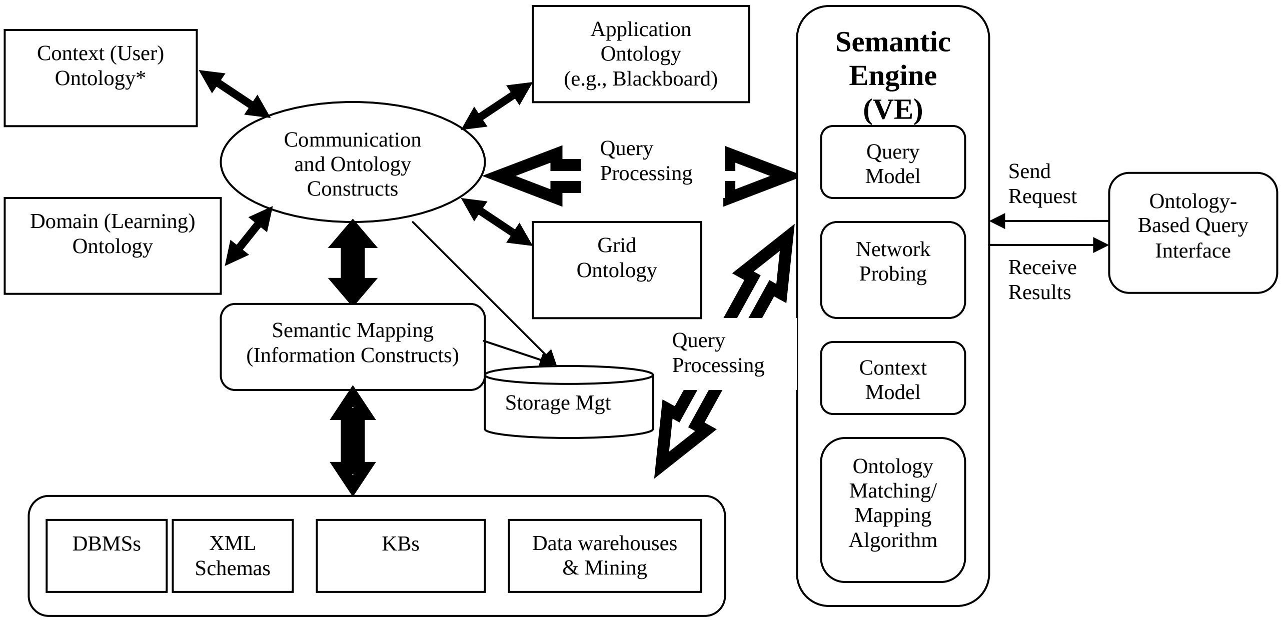

Figure 2: An Illustration of Collaborative System Architecture

The ontology matching algorithm is used to find semantic correspondence between ontologies. The matching algorithm should make a distinction between equivalent and similar ontologies where the latter case is almost impossible. As long as the ontologies of the source and destination are similar, the resulting knowledge will be similar as well. However, unless the source and destinations are clones, the transferred knowledge will never be the same. This is why there will always be some loss in translation in knowledge management, and the system needs to plan for it by anticipating the "spin" imposed by the destination and attempting to minimize it. The closer the ontologies are aligned, the more efficient the knowledge transfer will be.

3.3 Information Layer: An Integrated view of Information Nodes

The Web currently operates on the data layer when communicating with systems. The Semantic Web, by using RDF and RDFS, promises to upgrade the web from the data layer to the information layer (IL). Within the Semantic Web, meaning is transferable and information is reusable across the web. The Semantic web uses OWL to define inference rules and object-oriented taxonomies, which allow for distribution and inheritance of properties and meaning. However, this does not enable the semantic web with knowledge as it does not define systems for creating and executing the ontologies nor does it capture knowledge from the information. As mentioned earlier, this paper does not attempt to address the architecture of the semantic web, yet it augments it to create a holistic system that recognizes human-like functions.

The IL represents data and information storage. It is a semantic representation of data structure and its corresponding relationships, known as metadata. The database management systems, knowledge bases, RDF, and information sets (e.g., XML) are examples of the contents of the Information Layer.

Information items can be hosted in various databases or RDF tagged documents on the Semantic Web. The IL can also have intelligence functions that perform transformation of information into other information. IL can include simple inference rules, categorization based on the meaning, normalization, semantic and syntactic formatting, summarization, etc.

3.4 Context Modeling*

The proposed framework incorporates knowledge management and context awareness in the global network. While this model is not the only solution, the use of ontologies will evolve in the cooperation and e-learning systems. Once the ontologies are specified, they can be translated into functions and constructs, forming a communication framework, which coordinates and integrates all the ontologies as (shown in Figure 2. The mapping of multiple ontologies is accomplished using Java or OWL. The developer or user may combine different ontologies to create new entities or presentation that better represent the user specifications (i.e., contextual references). However, when dealing with two or more ontologies, one should define two different communication constructs: Internalization and Externalization constructs. Internalization constructs deals with the properties and characteristics of the user entities, entity instantiation and analysis, internal states, relationships between entities, and operations of entities within an ontology (i.e., contextual references); whereas externalization addresses the external relations and correspondence as well as translation between two or more ontologies and the external world. The Contextual references provide much granular classification of entities at both the internal and external environments, which are situational, empirical, interpretive, and subjective. The contextual references are captured by the Context Model, as described below.

Domain Ontology contains information specific to its knowledge domain(s). Each KN monitors user activities and registers user preferences in its *User Ontology* base system. Preference specifications allow users to access information customized to their particular usage. This practice avoids irrelevant or redundant information to be returned when requesting information

Context modeling addresses the issues of how to represent the contextual information in a way that can help bridge the gap between entity presentation and structure and the external world. Particularly, an object-oriented information model should be adopted in conjunction with Java Internet programming to generate useful information about the user interaction and classifications (categorizations of objects and membership types) to influence the meaning of the entities. Object-oriented models support the coordination and inter-relationships of data, metadata schemas, and interface to the data, thus enabling a transparent creation of knowledge from information by projecting data into problem specific virtual environments (e.g., views, VE). Relevant learning objects for a given peer context can be computed using a similarity measure between the current peer context and the ontological metadata of learning objects. This matching service has to take into account items such as:

- Current status
- Learning history and goal
- Skill drafting and assessment
- Social settings and structure
- Preferences & interests
- Entity interpretation

The most challenging and cautious aspect of shared knowledge modeling is the standardization of terminologies (vocabularies) and the elimination of definition ambiguity. This model brings forward a relaxed contextual framework that facilitates an evolutionary and reactionary ontological structure that is adaptive to its knowledge host.

4. Ontology Constructs

The following is the definition of the function parameters and outputs for each of the constructs defined at each of the layers already discussed in this paper.

4.1 Information Constructs

- Data mining function: $f_{d-m}^{DATA-INFO}$ *([target_media], [schemas], [query])* $\rightarrow I_{d-m}$, where I_{d-m} is information components of an information document, such as an XML document, RDF document, or a DB tuple or report component. $f_{d-m}^{DATA-INFO}$ takes an RDFS, XML Schema, or DB schema which defines meaning and scavenges the accessible media for data that could be encapsulated with that meaning.
- Semantic Web services: f_{seman}^{INFO} *([target_resource (web address)])* $\rightarrow$ *[*I_{seman}*, RDFS_schema, OWL_ontology]*, where I_{seman} is an RDF document; the *RDFS_schema* is the architecture of the meaning of the document, and the *OWL_ontology* is context of that meaning a network of its inter-relatedness to other meanings. The *OWL_ontology* is the knowledge enabling aspect.
- Inference rules / KB: f_{kb}^{INFO} *([KB_references],[information_set])* $\rightarrow I_{kb}$, where I_{kb} is the inferred set of information.

- Normalization: f_{norm}^{INFO} *([information_set], [related_ontology])* → I_{norm}, where I_{norm} is the normalized *information_set*. Normalization eliminates redundancy and useless components in the information set, based on the related ontology.

4.2 Ontology Constructs:

- Populating ontologies / knowledge agents: f_{kagent}^{KNOW} *([information_set], [ontology], [query])* → K_{kagent}, where K_{kagent} is knowledge for the query that is created by applying the information set into the referenced ontology.

4.3 Communication Constructs:

- Ontology creation and management: f_{ontoD}^{REASON} *([existing knowledge(within existing ontology set)], [problem_instance])* → *new_ontology,* where *new_ontology* is a new ontology created for the problem at hand, by adding or modifying ontology constructs from the referenced ontologies, through a creative process, tending towards a resolution of the problem.

5. CONCLUSION

The proposed model aims to improve information & knowledge sharing by bringing together the semantic web with collaborative systems, creating an environment whereby information is classified and delivered to participants according to their needs and more specific parameters. Future work will release system requirements and specifications, as well as interface prototypes, that will depict in more details the design and specifications of each system layer presented in this paper.

6. REFERENCES

[1] Boer, N.I. et al. An activity theory approach for studying the situatedness of knowledge sharing, *Proceedings of the Hawaii International Conference on System Sciences (HICSS-35'02), Hawaii*. Vol. 3, p. 90, 2002

[2] Chen, S., Alahakoon, D., Indrawan, M., Background Knowledge Driven Ontology Discovery, *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, pp. 202-207, 2005.

[3] Chen, H., Wu, Z., , Xu, J. KB-Grid: Enabling Knowledge Sharing on the Semantic Web, *International Workshop on Challenges of Large Applications in Distributed Environments*, pp. 70, 2003.

[4] Fernandes, A., Maria de C. Moura, A., Porto, F. An Ontology-Based Approach for Organizing, Sharing, and Querying Knowledge Objects on the Web, *14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, pp. 604, 2003.

[5] Deeb, K., A Context-based Ontological Structure for Knowledge Sharing and Customization, *The 2006 International Symposium on Collaborative Technologies and Systems, IEEE, Computer Society***,** 2006.