Knowledge Technology through Functional Layered Intelligence

Martin Dimkovski1, Kevin Deeb1
1 Barry University
11300 NE 2nd Ave., Miami Shores, FL, 33161
{mdimkovski, kdeeb}@mail.barry.edu

Abstract. The Internet and other various networks are growing into vast information pools and complex interconnection environments that are often unmanageable and overwhelming. Knowledge technology has been developing intensely at the turn of the century as the solution, however with limited success. The concepts such as intelligence, knowledge, information, and data are still amorphous. Existing efforts lack a clear overarching architecture model, which has led to a lack of coordination and interoperability. This paper introduces an architecture reference model, known as Functional Layered Intelligence (FLI) composed of intelligence functional layers, constructs, and functions. The FLI model, inspired by the OSI networking model, provides functional differentiation and interconnectedness. Its main contribution is that is allows for interoperability of currently competing computational models. In this paper, Knowledge technology is then presented as a subset of the FLI model.

Where is the Life we have lost in living?
Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?

1. Introduction

Due to the vast amount of information traversing the global networks, the proliferation of data stored in various information systems, and the increased complexity of the interconnectedness of all systems, we are facing an intensifying problem with the complexity, value, and accessibility of information. Companies are struggling with the massive amounts of accumulated information amongst various information systems and the lack of interoperability between them. End users are struggling with imprecise web searches and navigation through piles of information. For decades now, there have been many ongoing efforts to combat these issues through knowledge technology but these attempts have yielded very limited success [26].

Hai Zhuge, the chief scientist of China’s Knowledge Grid projects, identifies four limiting characteristics of the Internet and the WWW: rapid expansion of resources
and users, micro-structured and macro-less organization of resources, inequality of information, distribution, and machine-unreadable semantics, which cause "difficulty in accurate, effective, efficient, and safe use of globally distributed resources." [28]. The end result is a network of rapidly growing islands of information, which do not fall under any standard governance and universally effective translation systems.

The interrelationship of data, information, and knowledge is an existing hierarchy in organizational sciences describing concepts in human mind, referred to as DIKW (Data – Information – Knowledge – Wisdom), and also known as knowledge hierarchy, information hierarchy, knowledge pyramid, and other synonyms [1]. However, the hierarchy remains as amorphous as its first published reference in T.S. Eliot’s poetry [9]. Today, the same hierarchy is used in information sciences, knowledge management, organizational theory, and other sciences that work with concepts of human perception and cognition. Its amorphous structure is sometimes altered, such as by Ackoff who adds Understanding between Knowledge and Wisdom [1], or Zeleny, who adds Enlightenment after Wisdom [25]. Still today, the concepts and the transitions between the elements of DIKW remain intangible.

Knowledge is often de-contextualized, reified, or approached as an attribute to information. As Ikujiro Nonaka [16] points out, Western theories on knowledge “lack the view on the fundamentals of epistemology: what is knowledge, the nature of knowledge, and what constitutes learning. They are not clear about how the knowledge is captured, created, leveraged, and disseminated.” [17]. In effect, this research brings forth the distinction and co-relationship between information and knowledge and defines a Knowledge Technology (KT) architecture, which is modeled after the DIKW concepts and using the following FLI model, thus providing an architectural model for intelligence functions and constructs.

2. KT Today

The interconnectivity of the global network and the Internet has presented unprecedented opportunities and challenges, and has thus stirred up considerable research projects in KT. The overall aim of these projects is to improve the quality of information services and usability through coordinated and structured KT systems. Nonetheless, as Gräther notes, KT is still in its infancy, as "Current approaches to knowledge sharing communities like recommender systems or shared ontologies often suffer from an imbalance of effort versus benefit from the individual point of view" [12]. To understand the limitation we first have to become familiar with the major trends related to KT.

W3C’s (World Wide Web Consortium) with its initiative for the “Semantic Web” is a major driving force in these developments. As described by W3C, "the Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [4]. The Semantic Web concepts have been manifested in a system of standards and limited sample applications, which is structured in operational layers. The first and base layer is defined as W3C’s RDF (Resource Description Framework), which is a metadata language for representing information architecture. By using meaning description tags,
RDF enables the reusability of information across the Web, similarly to how XML enables the reusability of data within applications [14]. 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 is described using a separate schema description language called RDFS, which is the language for the second operational layer. There are extensive schema repositories on the web, such as http://www.schemaweb.org.

RDF is a universal tagging system that creates metadata using triples as [subject → predicate → object]. However, it lacks a mechanism for reasoning specifications [3][18]. While RDF creates information, it does not correlate it to create knowledge. It is designed to enable resource discovery queries on the Semantic Web. Automated Web agents can access the RDF to perform reasoning and knowledge creation.

The first two operational layers, RDF and RDFS, define the semantic labeling system. The actual definition of the meaning and semantic co-relations are concerted on the third and last layer through ontologies using W3C’s OWL (Web Ontology Language). Ontologies define inference rules and object-oriented taxonomies, which allow for distribution and inheritance of properties and meaning. Some of the functions of OWL are determining equality between classes, cardinality and relationships, describing properties as transitive, symmetric relationships, intersections of classes, unions, disjointness, etc. In essence, OWL directly aids the automated Web agents to assist in knowledge creation and reasoning specifications.

There are many design proposals, as well as ongoing research projects, to build KT on top of the Semantic Web. The KT designs usually involve some type of a directory that points to usable ontologies, an ontology server, a semantic browser, and system protocols [5][20][11]. There are also other KT efforts outside of the Semantic Web domain, using other interconnection environments. For example, there is much effort in utilizing grid architecture to achieve a form of a distributive KT system. This synergy is often referred to as the semantic grid. A prominent example is CKG, China’s Knowledge Grid Project [6][29][26][27].

CKG has released SVEGA-KG of version 2.0, which provides a tool to manually insert knowledge into local servers of the grid. It also provides a uniform and a universal resource view across the grid enabling distributed knowledge sharing and management. In version 3.0, they are hoping to incorporate intelligence to their KT software, such as automatic knowledge acquisition, reasoning, problem-solving, and decision making [28]. Another effort for enabling semantic grids is the Open Grid Services Architecture (OGSA) initiative by the Globus Alliance. The Globus Alliance is an organization dedicated to developing grid technologies.

Most of the KT projects and research efforts are related to knowledge bases (KBs) when it comes to ontology operations. The KBs are driven by inference rules that are used for selection, construction, management, and sharing of the ontologies. During the 1990s, DARPA was immensely involved in developing KB technologies, in particular through its Knowledge Sharing Effort (KSE), which produced standards such as KIF (Knowledge Interchange Format) and KQML (Knowledge Query and Manipulation Language) [23]. High Performance Knowledge-Based Processing (HPLB) and Rapid Knowledge Formation (RKF) were two other DARPA sponsored projects on formalist-reductionist knowledge bases.
The biggest and probably the most developed KB is the CYC project, by Cycorp, Inc. CYC’s functional operations are performed in first-order logic and organized in thousands of contexts [7]. CYC has some very unique applications in heterogeneous database integration, intelligent searches, WWW information retrieval, and natural language understanding.

KBs and inference rules introduce a formalist-reductionist dimension into whichever system they are applied. Using this approach in KT is not necessarily a wrong choice, but it may be limiting. The following is a review of some of the apparent and hypothesized limitations of the KT projects today.

3. KT Limitations

All of the present KT projects face the challenge of interoperability between ontologies, particularly due to the differences in the administrative domains and ontology definition languages. The core problem is that there is no predictability for the exact reductionist approach that was used to encode a particular referenced ontology, yet the nature of formalism requires exact syntactic and semantic matches with the ontology that is referencing. Even if we deal with existing systems for ontology conversion, exact rules defining allowable deviations are still required. The mix of formalism and reductionism seems to impose some inherent limitations in KT.

Some projects attempt to consolidate the heterogeneity of ontologies by correlating them and creating composite ontologies using articulation rules. It can be argued that this may simply shift the problem of coordinating ontologies to coordinating articulation rules. One of the prominent projects in this area was Stanford’s SKC (Scalable Knowledge Composition). The project achieved semi-automatic functionality since the human element was proven necessary to resolve conflicts in formalism mismatching. The system was able to learn from the human actions and repeat it. The project seems to have ended in year 2000.

DARPA’s HPKB project had ended in year 2000 and was replaced with RKF (1999-2003) [10], because HPKB proved that “Growing KBs by unassisted human “brainpower” was too slow, too expensive, and too prone to error.” This led to RKF, which focused on "allowing distributed teams of subject matter experts to quickly and easily build, maintain, and use knowledge bases" [10]. DARPA has been on the forefront of technology, yet today they seem to be less involved in further developing KB and KT technologies. Among the current public projects of DARPA, only the Real-World Reasoning (REAL) Project seems to be involved [8].

Semantic Web’s OWL native and abstract knowledge encoding capability has turned out to be too complex for programmability in formalist-reductionist systems. "It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full" [14], due to the complexity of the logic required. This is the reason W3C created two lighter and programmable versions, called OWL DL and OWL Lite. As W3C director Tim Berners-Lee points out: "Knowledge representation … is clearly a good idea, and some very nice demonstrations exist, but it has not yet changed the world." [4]. According to W3C, the Semantic Web is waiting for software agents that will perform interoperability. This, evidently, goes back to the
problem with ontology interoperability initially posed and the formalist-reductionist limitations.

One of the ways CKG is addressing the formalist-reductionist boundaries is using a process known as normalized resource organization. This process seeks to normalize semantic forms in attempt to eliminate redundancy, disorders, and useless aspects [28]. Their Resource Space Model (RSM), however, requires a "uniform viewpoint on resource partition", which brings back some of the formalist-functionalist limitations [27]. Zhuge also points out that the necessary, and currently lacking, interconnection environment, needs to have “organic characteristics such as self-protection, self-healing, fault tolerance, dynamic adaptation, self-replication, self-motivation, and self-fueling." [29]. These organic characteristics however are attributes inherently lacking in formalist-reductionist systems.

CYC has been growing for 21 years with hundreds of people building the knowledge base. It is an expert system that is useful for common sense knowledge; nonetheless, it is still far from being capable of effective KT. Additionally, CYC’s deduction reasoning is limited in depth iterations.

<table>
<thead>
<tr>
<th>Year</th>
<th>1998</th>
<th>1999</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE publications with &quot;knowledge Base(s)&quot; in the title</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>IEEE publications with &quot;knowledge&quot; in the title (DM 10)</td>
<td>7.8</td>
<td>13.6</td>
<td>14</td>
<td>16</td>
<td>17.2</td>
<td>15.4</td>
<td>16.7</td>
<td>14.8</td>
</tr>
<tr>
<td>CYC media coverage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>China Knowledge Grid publications</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>10</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>W3C Semantic Web presentations on workshops</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W3C Semantic Web publications</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Stanford KSL publications</td>
<td>30</td>
<td>17</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

![Figure 1](image)

Figure 1 shows the frequency of publications in the areas of KT by some selected renowned sources. The graph shows a general decrease in activity after a peak between 2001 and 2003. The interesting factor is that they all have different results as

---

1 The count is derived from available information.
2 The 2005 count is estimated assuming the same frequency throughout the year, as until now.
3 The graph is represented using a stacked line model displaying the trend of the contributions of each source to the total publication frequency over time.
well as diverse funding sources. We can speculate that this is due to discipline-wide limitations and scalability issues reached by KT research. Most KT applications require a human element to resolve the formalistic contentions, as well as to outline the reductionist process. Predictable and programmable reductionism, combined with formalism, seems to enact effective KT only within well-defined, limited, and predictable environments. Of course, it could also be due to other factors, such as the burst of the market bubble in IT.

The success of the limited KT applications should not be underestimated. They have proven to possess excellent and irreplaceable applicability. Also, the formalist-reductionist systems cannot be pushed aside, because they provide critical functional benefits. The following presents the FLI architecture that combines the formalist-reductionist models with other computational models.

4. FLI Defined

FLI is an architecture reference model, not a new system or a new set of protocols. FLI is not addressing a lack of protocols or designs, but the unstructured and uncoordinated development of the KT applications without an overarching functional intelligence paradigm. This paper does not present any specific implementation following FLI but only the architectural reference model itself.

The goal of FLI is not to model the human-being in all its intelligence and other mental functions, though it does not preclude this scenario. Instead, FLI is designed to enable integration of human-like intelligence functions into technological systems in order to address desired functionalities. The goal of FLI is to enable human-like functions, such as deduction, induction, inference, understanding, or wisdom for systems like the Internet, the Web, personal assistance devices, expert systems, or monitoring systems, among others. FLI is designed for practical integration into today's communication and computing infrastructure.

FLI is a layered architecture reference model that is inspired by the OSI layered networking model. FLI defines stand-alone functional layers using the conceptual domains: Data, Information, Knowledge, and Reasoning. The system process is executed upwards or downwards through the layers, where each layer calls upon the functionalities of its adjacent layer(s). Each layer does not have to understand the domain of its neighboring layer. A layer may pose a request to a neighboring layer through a function with parameters that are native to its own domain, and gets results back in its native language.

Each layer hosts its corresponding DIKW concept, such as information or knowledge. The only exception is the Reasoning layer, which combines understanding (for which it can be also argued that it belongs to the Knowledge layer) with wisdom. These concepts are referenced by FLI as intelligence constructs. The following depicts two FLI based systems and their communication relationships:
Fig. 2. Two FLI systems communicating through peer level communication

FLI is an architecture reference model for intelligence functionality between systems. It builds upon the concepts introduced in DIKW by creating computational functions at each layer. The purpose of the functions within FLI is to enable the intelligence constructs to be translated across the layers or managed within them. The actual physical communication happens on the Data layer. Once data is transmitted to a peer intelligent system, it is then reconstructed back into its original, higher level construct. When the layers on a system communicate downwards, they take their native construct and deconstruct it through the functions of their layer into a form that the underlying layer is capable of carrying. They convert it to a combination of the lower intelligence constructs, together with a deconstruction “blueprint”. At the same token, when communication on the receiving system moves upwards in the layers, the lower layers deliver their intelligence constructs, along with the agreed “blueprint”, to the upper layer where the constructs and the blueprint is used to construct the higher level intelligence construct. During this process, each layer of the system appears as communicating directly with its corresponding peer level at the peer system. Hence, Information is communicating with Information, Knowledge with Knowledge, etc. FLI can be expanded in the future if more layers are warranted. The layers can also be expanded to include more constructs if necessary.

*Intelligence functions* are the processes that operate within the layers or across (in-between) layers, and perform the creation, transformations and management of the intelligence constructs. They are responsible for the deconstructing and reconstructing of the constructs, as well as for negotiating and executing the blueprints of those constructions. An example of an already existing intelligence function that operates between the Data and the Information layers are databases. Databases, particularly object-oriented ones, are abstraction layers over raw data that maintains inter-relationships of the data, together with metadata schemas and interfaces to the underlying data. By projecting data into meaning-specific environments (views, reports, etc), databases transparently create information out of data.

Applying information to ontology and constructing knowledge is an intelligence function between the Information and the Knowledge layers. The ontology is the abstract blueprint in this scenario. Nonaka describes this segment of FLI indirectly in an interview: “In very simple terms, information is the flow, and knowledge is the stock. Information is the flow of a message, while knowledge is created by accumulating information. Thus, information is a necessary medium or material for eliciting and constructing knowledge.” [17].

The Reasoning layer, together with its relating intelligence functions, is the most undeveloped layer today. This layer extracts knowledge from the underlying layer, and constructs reasoning. What is conventionally known as Artificial Intelligence (AI)
relates to the level. More specifically, this layer is related to induction, deduction, understanding, wisdom, and other higher level intelligence ability.

FLI enables designers to provide multiple implementations of an intelligence function, even using separate computational models. It is up to the layer and the parameters of the request to decide which intelligence function to call for translation or management. The functions can be addressable by choice, downwards and upwards. Through this ability, FLI creates an architecture where competing computational models can co-exist and offer functionalities to each other.

<table>
<thead>
<tr>
<th>Data</th>
<th>Information</th>
<th>Knowledge</th>
<th>Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMS</td>
<td>XML Schemas</td>
<td>Data Mining</td>
<td>Optical Recognition</td>
</tr>
<tr>
<td>Traditional Web Servers</td>
<td>File Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formatting</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 3. FLI architectural model with sample intelligence layers, functions, and constructs](image)

FLI can also be used for modeling intelligence functions within a single system that is not communicating with other systems. A layer may call the functions of a neighboring layer and receive the results itself. However this scenario can always be viewed as the system communicating with itself.

Today's traditional networks, including the current web, operate on the Data layer only when communicating between systems. The Semantic Web, by using RDF and RDFS, promises to upgrade the web from the Data layer to the Information layer. Within the Semantic Web, meaning is transferable and information is reusable across the web.

Adding OWL into this mix enables the Semantic Web for supporting blueprints for knowledge. However is does not move the Semantic Web to the Knowledge layer, because it does not define systems for creating and executing the blueprint (ontologies), nor does it capture knowledge from the information. This is precisely why the Semantic Web is waiting on the software agents as mentioned above, which will unlock its full potential.

---

4 Rectangular shapes denote cross-over intelligent functions, whereas oval shapes denote in-layer intelligent functions.
5. KT through FLI: Ontologies and Knowledge

Knowledge is more than passive sets of information that share a vocabulary. It is the product of their co-relationships and the network they form. Knowledge becomes active when that network is put in use. Ontology, on the other hand, should not be viewed as a set of formal logic statements, even less as a vocabulary, but as a network of meaning. The formal logic is just one way of describing the interconnections of the network. "Seeing knowledge from a formal logic perspective forces the focus onto the individual predicates, the building blocks of formal logic. Conceiving of knowledge as a network provides a comprehensive or gestalt view, which facilitates seeing the forest rather than the trees." [15]

The ontology, a network of meaning, is a system where information co-relates into knowledge. It is a symbolic form of computation, unlike conventional procedural computation which is attributed with formalism and reductionism. Of course, we can attempt to use a formalist-reductionist model to produce symbolic computation by breaking down meaning and formalizing its components, and then applying procedural computation. Discussing the feasibility of this approach is beyond the scope of this paper, but as reviewed previously, it has been the dominant approach in KT so far and has not yet led to any promising results. Using FLI, we have the capability to compliment this approach with alternative methods of computation.

Knowledge is indivisible from its ontology. Ontologies are not just descriptors or helpful tools for knowledge, but they are the architecture of knowledge. They map out the co-relation of structural elements of knowledge. Knowledge itself is defined as a dynamic “multifaceted concept with multilayered meanings” [16], and ontologies provide the multifaceted environment which, when populated with information into a network of meaning, gives birth to knowledge. "After all, one can not observe what knowledge is, but only to some extent what knowledge does." [2]

Knowledge does not exist outside of its ontology, even during knowledge transport. The moment it is taken out of the ontology, (for transport, transformation, etc.) it is immediately de-contextualized and broken down simply to information, being moved from the Knowledge layer to the Information layer. When these de-contextualized pieces are delivered to the destination (over the Data layer), they get injected into the new host ontology and become part of a new knowledge, as they are processed upwards through the receiving layers. The new copy will be as authentic as the destination ontology is to the source one. Going back to Nonaka, “Information is the flow of a message, while knowledge is created by accumulating information. Thus, information is a necessary medium or material for eliciting and constructing knowledge.” [17]. FLI outlines this through the transitions between the Information and the Knowledge layers.

In the process of knowledge transfer, unless the source and destinations are clones, the transferred knowledge will never be the same. Cloning ontologies is as difficult as cloning anything. This is why some loss in knowledge transfers should always be accounted for, and the system needs to plan for it by anticipating the “spin” imposed by the destination and attempting to minimize it. Error checking on the transmitted knowledge can be part of the communication protocol between the FLI systems, similar to error checking as part of the Ethernet communication protocol in the OSI layered model. The closer the ontologies are aligned, the more efficient the knowledge
transfer would be. Likewise, if the destination ontology has a serious anomaly, effective knowledge transfer will not succeed, regardless of how well function is executed.

6. Illustration

Let’s consider a sample KT operation in a FLI based system. It would start with a perception process (“looking”). This process is done in the Data layer. Meaning can then be associated to the perceived content based on the environment in which the perception process is executed. The technology for meaning assignment is considerably developed today. The processes for meaning assignment are the intelligence functions between the Data and the Information layer. They include: data mining, optical and voice recognition, database data definition languages, etc. The Data layer can also have intelligence functions that operate within the layer itself and perform transformation of data, such as compression, formatting, etc.

Once meaning is assigned to the data, information is created in the Information layer. This information can be hosted in various databases (relational, object-oriented, XML), or RDF tagged documents on the Semantic Web. The Information layer can also have intelligence functions that perform transformation of information into other information. They can include simple inference rules, categorization based on the meaning, normalization, semantic and syntactic formatting, summarization, etc.

When the information is ready at the Information layer, the Knowledge layer can now, using KT, generate the necessary ontologies. The components for the ontology could be selected using knowledge bases, directory servers, associations, etc. Ontology components include inference rules, classes of meaning, object relationships, inheritance rules, set rules, etc. The ontology creation process resides fully in the Knowledge layer. The ontology it creates, however, becomes a connection between the Information and the Knowledge layer.

In order to create the actual ontology, the selected components have to be injected into an interconnection environment and correlated into a network of meaning. This is the part that mostly requires the organic characteristics. Once the ontology is created and tapped into its related information sources at the Information layer, it becomes knowledge capable for the particular request domain. Executing a request from that domain produces a knowledgeable response.

7. The Network-Centric Approach for the Knowledge Layer

This paper argues that the necessary organic characteristics for KT can be most optimally achieved using a network-centric based Knowledge layer, and also one not limited only to formalist-reductionist methods. KT built into a network-centric model would enable it to reach over all the information available through the network resources. It can be overlaid on an information layer that is functionally ready to provide information services to it, without having to build KT layer on legacy member resources. The KT potential of a network-centric model lies in its abstraction layer and its overseeing power.
The qualifying attribute for a network-centric model is the positioning of the network in relation to the user node. In a network-centric paradigm, the network (viewed from the “port on the wall”) is not just the relatively “dumb” connecting medium for the “smart” end-user nodes. Instead it integrates the user node into a distributed “smart” system. The “smartness” of the user node and its involvement in this model can vary. The user request is performed by the network, and servicing of the request through the network’s distributed resources is transparent to the user. The network becomes the point of service, not a server across the network.

Utilizing a network-centric model at the Knowledge layer also provides an environment where the ontologies necessary for KT can be more represented and hosted more natively. The network-centric model manages inter-connections of its resources. This is more native to ontologies if it is approached as networks of meaning, as discussed previously. The ontology can then be natively presented as a interconnection environment. The network centric model has already been shared by many existing KT projects, particularly those that are based on semantic grids. However, grids are not the only type of network-centric technology.

The current implementations of the Knowledge layer fails to create the “interconnection environment” needed to generate on-demand ontologies and manage their dynamics and transformations. Many pre-requisites for the interconnection environment are not clear themselves. For example, it is still not clear how the semantics of the complex ontology inter-relations (inheritances and combinations) in the Semantic Web will look like [30]. As Zhuge points out, the interconnection environment will need to have organic characteristics such as self-protection, self-healing, fault tolerance, dynamic adaptation, self-replication, self-motivation, and self-fueling [29]. Many scientists also point out that “the relationships within a simulated knowledge conceptual network cannot be handcrafted, but instead must be self-organizing.”[15]. The KT scientific community has acknowledged the needs for these organic characteristics in order to overcome scalability and limitations of semi-automation. FLI enables us to integrate methods that address organic characteristics natively; not necessarily as a replacement to the existing ones, but in combination. For example, fuzzy logic or probabilistic systems such as Bayesian networks can be used to address fault tolerance and dynamic adaptation. Emergent computing methods such as neural networks can expand the computational model beyond the formalism and reductionism.

8. Conclusion

The specifics of the possible implementation are each topic of a separate paper. The goal here is to introduce an architectural reference model that can enable development of KT systems, out of their apparent halt right now, or their seemingly inevitable conflict with formalism and reductionism.

We need a solution for "knowledge [that] will evolve and endure throughout the life of the human race rather than the life of any individual." [29] KT is becoming increasingly important, especially with the increasing overload and overhead in information processing. Current efforts in KT have reached serious limitations in scalability and effectiveness. They seem to be limited to explicit application domains. There
is also no overarching architecture model within which these developments are going on, preventing optimal inter-operability between their efforts.

FLI can serve this purpose by providing an architecture reference model. Within FLI intelligence, functions can be implemented using various technology as well as computational methods. FLI also divides the responsibilities over the various intelligence constructs.

Viewed through FLI, most IT systems today are operating on the Data layer. A lot of effort and success is going on for the Information layer. However there are serious challenges in the area of the Knowledge layer. KT is a functionality of the Knowledge layer. FLI enables KT to utilize alternative technologies and computational methods such as fuzzy logic, probabilistic systems, Emergent computing, neural networks, and others. These alternative approaches can provide the necessary organic characteristics and overcome the limitations of the formalist-reductionist approach.

9. References