Chapter 1

The Current State of KBS

Knowledge-based systems are changing our society, but they are hardly a fresh innovation. Educational systems that preserve and share the archived knowledge of the past have been around since the very first student was given a lesson by a wise elder. In modern times, we spend a sizable portion of our lives in learning institutions working hard to attain varying levels of education. We receive diplomas as recognition of our efforts, and our worth on the job market is measured by our degrees and experience. We value the gathering of knowledge above almost any other human pursuit, and we have always sought faster and more efficient ways of collecting and utilizing it. Capturing knowledge is simply nothing new.

What is new are computers. Fantastic machines which can do calculations and step through processes faster than any human, and also mimic our ability to do logical deductive reasoning. Because of this capability, knowledge captured in a form understandable by computers can also be used to perform complex logical analysis by computers. Computing systems built with the intent to use captured knowledge to drive processes are called knowledge-based systems (KBS). We humans have to attend school for years to reach some level of understanding of the world we live in. Computers get theirs from us without any such effort or time investment. It all comes down to computers having a capacity to understand, not an ability to learn. KBS is about humans capturing knowledge and providing it in a computer-readable form.

Automated systems execute their tasks by running programs, which are written in a code understood by programmers and computers. All the knowledge required to run a program must be included in the code. When a program has to reflect a domain of knowledge, the programmer must possess a full understanding of that knowledge. If that knowledge is of a complexity that takes years of training to acquire, it can be nearly impossible for the best of programmers to write adequate code within a reasonable time frame. Consider a programmer challenged to write a program describing reactions in the knowledge domain of chemistry. Without a complete understanding of chemistry, the programmer has little chance of producing a worthwhile program without first gaining that understanding, but odds are the deadline for completing the program will occur long before the programmer can earn a full chemistry degree.

With the almost unimaginable speed of computers today and the vast amount of storage available, programming has quickly advanced beyond being a mere vehicle to automate business processes. Now computers are used for major research and tracking of events. To meet these new challenges, we don't need to just write bigger and better programs, we need to capture knowledge in such a fashion that programmers can use and build upon that knowledge without having to have a full understanding of it themselves.

This is where KBS comes in. KBS is not about programming knowledge. KBS implementations separate the encoding of knowledge from the coding used by programmers. Those that have the knowledge can encode it without needing to understand the complexities of programming, and the programmers do not need to have the depth of learning required to utilize the knowledge domain.

Storing knowledge separately from program code is not a new approach. In the last few decades, there have been many successful implementations of systems by capturing rules external from the program code. More recently, the advancement is to define ontologies containing the logic of knowledge. Provability of the logic can then be accomplished by automated reasoning. In very complex environments where no single person can comprehend the full knowledge of an environment, ontologies are becoming the norm to give computers the ability to mathematically prove the consistency of the structure of the encoded knowledge of ontologies.

There are thousands of very intelligent people worldwide who strongly believe in the value of KBS, and are proving its value with important implementations. Examples range all the way from finding a cure to cancer, to tracking the movement of everything on earth and in the universe.

1.1 Ontology and Information Technology

1.1.1 Conferences

Have you ever walked out on the street of a major city and noticed a crowd of people standing and looking up at the sky, while others on the street kept their heads down and continued moving on to their destination? If you have, did you look up or did you continue on your way? Perhaps you never even noticed the crowd looking up because you had your head down and were so focused on reaching your destination.

If your head is down, look up. There are many crowds forming and looking up at the potential of KBS. They see tremendous opportunity to help mankind by fighting disease, predicting natural disasters, and explaining our universe. It's truly worth pausing a moment to look up and see what they're seeing. If not, you will miss out on one of the greatest potential applications in the use of computers.

In 2011, I had the enlightening opportunity to attend IC3K 2011, International joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. This was the third meeting of this international group. The IC3K in preparation for this meeting received 429 paper submissions from 59 countries in all continents. Along with these papers discussing KBS, many went beyond theory by also presenting computer-based prototypes.

The Appendix **??** includes an overwhelming list of knowledge engineering conferences. Thousands of dedicated individuals on an international scale attend these conferences. They recognize the need to share and collaborate on the capturing and use of knowledge by building knowledge-based systems.

1.1.2 Standards

Defining and storing knowledge has presented a challenge to develop new languages. Today, computers need precision and clarity. They don't deal well with paradoxes. For example, a paradox attributed to the Greek mathematician Zeno is the proposal that a runner can never finish a race, because the runner always has to complete half the distance to the finish line first, then half the distance remaining from the midway point, then half the distance remaining from the next midway point, and so on and so on, forever whittling down the distance by halves. As there would always be another half to complete, the runner could never reach the end, since there would forever be some fraction of the distance remaining. We, of course, understand that Zeno was only illustrating a point and knew full well the runner would eventually cross the finish line no matter how the distance was sliced, but presented with this same nonsensical formula, a computer would endlessly halve the distance as instructed and agree that the runner could never reach the goal. Computers need languages that will allow them to consider illogical ideas without being tripped up by them.

I have not counted them all, but there were more than a few episodes of the original Star Trek series where Captain Kirk challenged a massive computer with a paradox and caused it to blow up. Even back in the 60's, people understood that computers are not good at handling a paradox, though in real life, a computer is highly unlikely to become so engaged trying to solve a paradox - or any problem, really - that its circuits heat up and explode. This, luckily for those of us who spend much of our lives sitting in front of a computer, was one prediction of the future that Star Trek got completely wrong.

The W3C (World Wide Web Consortium) has stepped up to the challenge of providing these new languages. They have categorized them in the general context of the Semantic Web. The idea behind the Semantic Web is simple. Data on the web should be easily accessible through automated means. Where HTML is designed for sharing information for humans, the Semantic Web is intended to share information with computers.

The structure selected to capture knowledge-based information is in the form of a graph. A graph is not a chart like a bar chart or a line chart. These charts are simply a form to visualize data. Graphs in semantics represent the structure of knowledge-based information, not the visualization.

Graph structure is not hierarchal nor is it relational. It is a network. It is not hierarchal because it does not have a top or a bottom. It is not relational because it does not have rows and columns as in a relational database.

The network structure has only two components, nodes and edges. Nodes represent things and edges represent the relations between things. Nodes are usually viewed as objects and edges are usually viewed as lines connecting the objects. A real life view of a graph resembles a network having no beginning and no end. It is simply a set of things connected by lines.

The term graph made up of nodes and edges is based in mathematics. Mathematicians have for hundreds of years recognized the nature of graphs and their application to real world challenges. They have defined proven algorithms we use every day. Some algorithms are used to find the shortest path through a network or how to connect nodes with the minimum cost to reach a goal.

GPS systems use the shortest path algorithm to find the route to a destination where roadway intersections are nodes and the roadways are edges. These algorithms are also used in business in many ways. In project management, shortest path is applied to find the critical path within a project. Nodes are the project activities and the edges are the dependencies of one activity upon another. It is used in logistics to find the best route for delivering products in a supply chain. The nodes are the providers, warehouses, and destinations. The edges are the routes available. In network design, the algorithms are used to find the minimum distance to connect multiple remote systems. The remote systems are the nodes and the connections are the edges.

The W3C first introduced the RDF (Resource Definition Framework) language to define things and relations. The language OWL (Ontology Web Language) was introduced later that built upon RDF and provided additional axiomatic capability. Both the RDF and OWL languages provide syntax to define nodes (things) and edges (properties). Things are defined using class structures supporting sub-classes, multiple inheritance, and equivalence. Properties are defined using axiomatic qualifications.

Both RDF and OWL also provide the syntax to declare individuals that exist within the structure defined. Individuals can be thought of as the data where the structure is the metadata (data about the data). The combination of the metadata and the data into single documents represents a complete component of knowledge. Having a mathematical base, RDF or OWL documents can be validated to be axiomatically sound. This is an important capability and will be covered later in more detail.

In addition to RDF and OWL, the W3C developed the language SPARQL (SPARQL Protocol and RDF Query Language) to provide the ability to query information stored within RDF. SPARQL provides a means for defining queries to retrieve individual data from RDF in a similar manner as using SQL to retrieve data from a relational database. The W3C also developed a language in which to define rules, SWRL (Semantic Web Rule Language). SWRL provides a means of externalizing rules that control events. The language syntax is appropriate for those that maintain business rules.

1.1.3 Ontology repositories

Sharing knowledge is the very noble driving force behind establishing repositories. It is a modern version of a public library. These libraries contain great works of art that attempt to describe the universe we live in. These attempts are based on scientific facts and on observations that may lead to facts.

All of those that participate in capturing knowledge and sharing are cultural heroes seeking to make life better for us all. They have devised these ontology repositories containing thousands of ontologies. Each of the repositories provides basic functions to add and update ontologies. The ontologies are usually organized within categories to make it easy to see the scope of the knowledge provided. Many offer search capability based on keywords.

In this section, three well known repositories are mentioned:

- OntoHub - Contains ontologies and other repositories

- Tones - A central location for ontologies that might be of use to tools developers for testing purposes

- BioPortal - A repository of biomedical ontologies

OntoHub OntoHub is described at `http://wiki.ontohub.org/index.php/Main_Page` as:

> Ontohub is an open ontology repository which supports organisation, collection, retrieval, development, mapping, translation, and evaluation of a wide array of ontologies formalised in diverse languages.
>
> Key features of Ontohub:
>
> - OntoHub is an ontology repository that supports the ontology development and maintenance along the whole ontology lifecycle.
> - publishing & retrieval: OntoHub is a free ontology repository that allows you to publish your ontology and find existing ontologies that are relevant for your work.
> - development: OntoHub supports ontology development. Since OntoHub is based on Git repositories, OntoHub supports ontology versioning, branching, and merging.
> - evaluation: OntoHub is designed to be a platform for ontology evaluation tools. The goal is to support all kinds of evaluation; including syntactic validation, best-practices evaluation, and regression testing.
> - multilingual: OntoHub supports a wide variety of languages and logics. The same ontology may exist in more than one language. OntoHub supports the automatic translation between languages.
> - open: OntoHub is based on open source software.

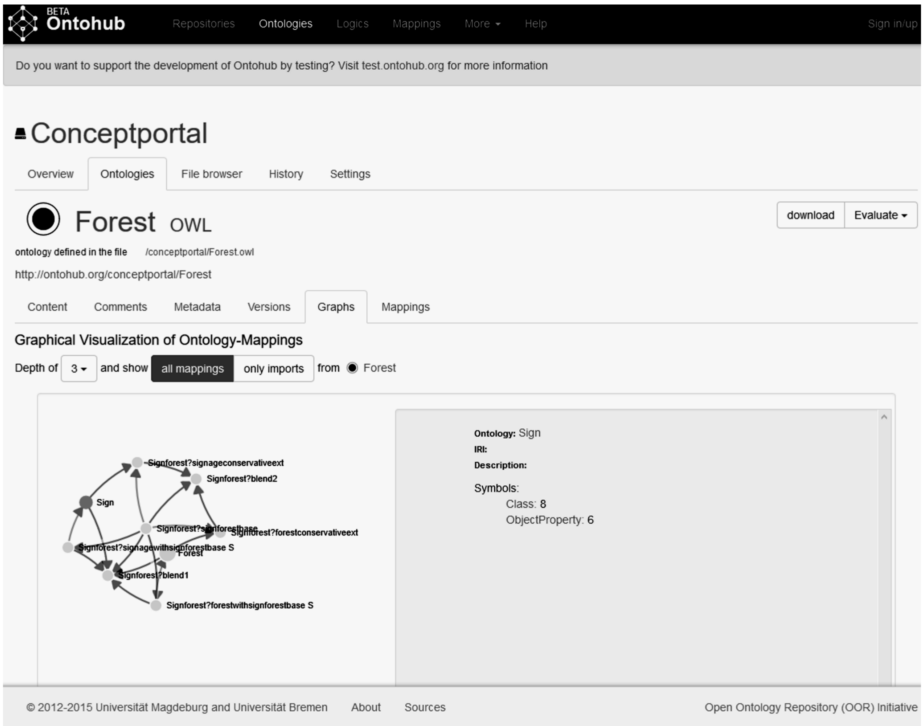
At the time of this writing, the OntoHub contained 2,950 ontologies and 56 repositories and organized into the following categories:

- Agriculture Forestry Fisheries Veterinary
- Arts and humanities
- Business administration and law
- Education
- Engineering manufacturing and construction
- Health and welfare
- Information and Communication Technology
- Natural science mathematics and statistics
- Services
- Social science journalism and information

- Space Time and Process

- Standard method and research technique

As an example, in the category of "Natural science mathematics and statistics" the repository called SpacePortal is listed in the sub-category of "Mathematics". It contains 63 annotation properties, 3,534 classes, 32 data properties, 1,226 individuals, and 326 object properties.



Do you want to support the development of Ontohub by testing? Visit test.ontohub.org for more information

Conceptportal

Forest OWL

ontology defined in the file /conceptportal/Forest.owl

http://ontohub.org/conceptportal/Forest

Graphical Visualization of Ontology-Mappings

Depth of 3 ▾ and show all mappings only imports from ● Forest

Signforest?signageconservativeext
Signforest?blend2
Sign
Signforest?signforestbase
Signforest?forestconservativeext
Signforest?signagewithsignforestbase S
Forest
Signforest?blend1
Signforest?forestwithsignforestbase S

Ontology: Sign
IRI:
Description:

Symbols:
Class: 8
ObjectProperty: 6

© 2012-2015 Universität Magdeburg and Universität Bremen About Sources Open Ontology Repository (OOR) Initiative

Figure 1.1: OntoHub ConceptPortal Graph View

Another example, the ConceptPortal that is described as *"populated with (common sense) ontologies in particular from the mathematics and music domains that shall be used to support and enrich the blending process"*. When selecting this repository, the OntoHub provides multiple views of the ontologies.The graph view is shown in Figure 1.1.

The graph view shows the classes and the relations between the classes. Figure 1.2 is a focus on the graph. The classes are nodes and the relations are lines. When the class "Sign" is selected as in Figure 1.1 the right side of the view shows the more detailed information about the class as shown in 1.3.

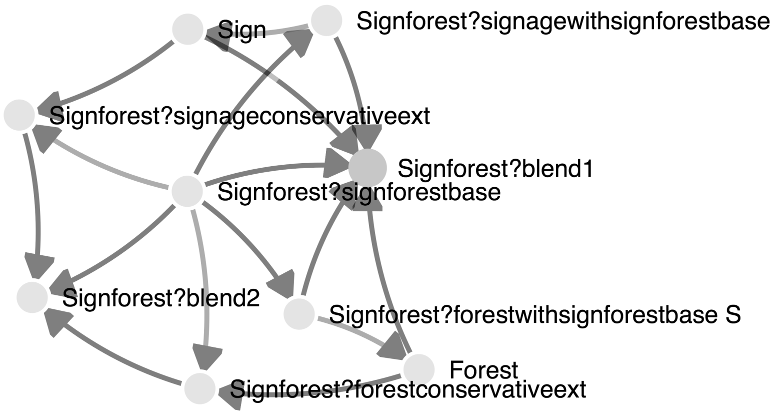
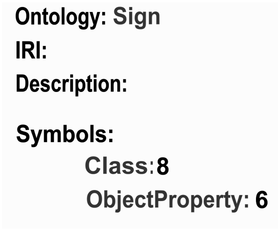

Sign
Signforest?signagewithsignforestbase
Signforest?signageconservativeext
Signforest?blend1
Signforest?signforestbase
Signforest?blend2
Signforest?forestwithsignforestbase S
Forest
Signforest?forestconservativeext

Figure 1.2: OntoHub ConceptPortal Graph


Ontology: Sign

IRI:

Description:

Symbols:

Class: 8

ObjectProperty: 6

Figure 1.3: OntoHub ConceptPortal Ontology

The OntoHub brings together many of the ontology repositories available on the web. In fact, the next two repositories are included in the OntoHub.

Tones The Tones repository is provided by the University of Manchester in the UK and supported by the Tones project. The University of Manchester is known for their work in ontology especially for the Protégé integrated development tool for OWL documents.

The Tones project is described at `http://www.tonesproject.org/` as:

> TONES (Thinking ONtologiES) is an Information Societies Technology 3-year STREP FET project financed within the European Union 6th Framework Programme under contract number FP6-7603.
>
> The aim of the project is study and develop automated reasoning techniques for both offline and online tasks associated with ontologies, either seen in isolation or as a community of interoperating systems, and devise methodologies for the deployment of such techniques, on the one hand in advanced tools supporting ontology design and management, and on the other hand in applications supporting software agents in operating with ontologies. This site contains use-

> ful informations about the TONES Consortium Organization, and about the ongoing and completed project activities.

The repository contains 219 ontologies that can be searched using filters.

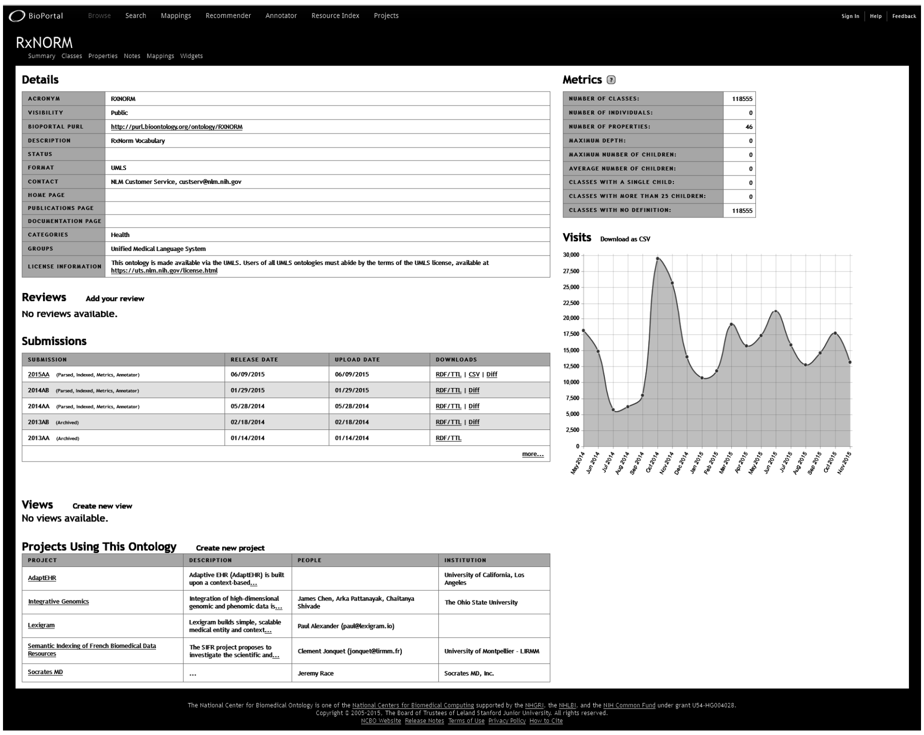
BioPortal BioPortal is described on the website (`http://www.bioontology.org/BioPortal`) as:

> THE NATIONAL CENTER FOR BIOMEDICAL ONTOLOGY BioPortal is an open repository of biomedical ontologies that provides access via Web browsers and Web services to ontologies. BioPortal supports multiple ontology formats. It includes the ability to browse, search and visualize ontologies as well as to comment on, and create mappings for ontologies. Any registered user can submit an ontology. The NCBO Annotator and NCBO Resource Index can also be accessed via BioPortal.
>
> BioPortal Features:
>
> - Browse, search and visualize ontologies.
> - Support for ontologies in multiple formats (OBO format, OWL, RDF, RRF Protégé frames, and LexGrid XML).
> - Add Notes: discuss the ontology structure and content, propose new terms or changes to terms, upload images as examples for terms. Notification of new Notes is RSS-enabled Notes can be browsed via BioPortal and are accessible via Web services.
> - Add Reviews: rate the ontology according to several criteria and describe your experience using the ontology.
> - Add Mappings: submit point-to-point mappings or upload bulk mappings created with external tools. Notification of new Mappings is RSS-enabled and Mappings can be browsed via BioPortal and accessed via Web services.
> - NCBO Annotator: The Annotator is a tool that tags free text with ontology terms. NCBO uses the Annotator to generate ontology annotations, creating an ontology index of these resources accessible via the NCBO Resource Index. The Annotator can be accessed through BioPortal or directly as a Web service. The annotation workflow is based on syntactic concept recognition (using the preferred name and synonyms for terms) and on a set of semantic expansion algorithms that leverage the ontology structure (e.g., is_a relations).
> - NCBO Resource Index: The NCBO Resource Index is a system for ontology based annotation and indexing of biomedical data; the key functionality of this system is to enable users to locate biomedical data linked via ontology terms. A set of annotations is generated automatically, using the NCBO Annotator, and presented in BioPortal. This service uses a concept recognizer (developed by the National Center for Integrative Biomedical Informatics, University of Michigan) to produce a set of annotations and expand them using ontology is_a relations.

- Web services: Documentation on all Web services and example code is available at: BioPortal Web services.

As an example, a query on the RxNORM ontology shows the results in Figure 1.4. The query results includes details, metrics, visits, reviews, submissions, views, and projects using this ontology.

What is outstanding is the number of visits and projects. Visits to this ontology reached a peak of nearly 30,000 in November of 2014. The projects are using the information to better understand medical imaging, integrated genomics, French biomedical data resources, and artificial-intelligence.



RxNORM

Details

ACRONYM	RXNORM
VISIBILITY	Public
BIOPORTAL PURL	http://purl.bioontology.org/ontology/RXNORM
DESCRIPTION	RxNorm Vocabulary
STATUS	
FORMAT	UMLS
CONTACT	NLM Customer Service, custserv@nlm.nih.gov
HOME PAGE	
PUBLICATIONS PAGE	
DOCUMENTATION PAGE	
CATEGORIES	Health
GROUPS	Unified Medical Language System
LICENSE INFORMATION	This ontology is made available via the UMLS. Users of all UMLS ontologies must abide by the terms of the UMLS license, available at https://uts.nlm.nih.gov/license.html

Metrics

NUMBER OF CLASSES:	118555
NUMBER OF INDIVIDUALS:	0
NUMBER OF PROPERTIES:	46
MAXIMUM DEPTH:	0
MAXIMUM NUMBER OF CHILDREN:	0
AVERAGE NUMBER OF CHILDREN:	0
CLASSES WITH A SINGLE CHILD:	0
CLASSES WITH MORE THAN 25 CHILDREN:	0
CLASSES WITH NO DEFINITION:	118555

Visits Download as CSV

Reviews Add your review

No reviews available.

Submissions

SUBMISSION	RELEASE DATE	UPLOAD DATE	DOWNLOADS
2015AA (Parsed, Indexed, Metrics, Annotator)	06/09/2015	06/09/2015	RDF/TTL | CSV | Diff
2014AB (Parsed, Indexed, Metrics, Annotator)	01/29/2015	01/29/2015	RDF/TTL | Diff
2014AA (Parsed, Indexed, Metrics, Annotator)	05/28/2014	05/28/2014	RDF/TTL | Diff
2013AB (Archived)	02/18/2014	02/18/2014	RDF/TTL | Diff
2013AA (Archived)	01/14/2014	01/14/2014	RDF/TTL
			more...

Views Create new view

No views available.

Projects Using This Ontology Create new project

PROJECT	DESCRIPTION	PEOPLE	INSTITUTION
AdaptEHR	Adaptive EHR (AdaptEHR) is built upon a context-based...		University of California, Los Angeles
Integrative Genomics	Integration of high-dimensional genomic and phenomic data is...	James Chen, Arka Pattanayak, Chaitanya Shivade	The Ohio State University
Lexigram	Lexigram builds simple, scalable medical entity and context...	Paul Alexander (paul@lexigram.io)	
Semantic Indexing of French Biomedical Data Resources	The SIFR project proposes to investigate the scientific and...	Clement Jonquet (jonquet@lirmm.fr)	University of Montpellier - LIRMM
Socrates MD	...	Jeremy Race	Socrates MD, Inc.

The National Center for Biomedical Ontology is one of the National Centers for Biomedical Computing supported by the NHGRI, the NHLBI, and the NIH Common Fund under grant U54-HG004028.
Copyright © 2005-2015, The Board of Trustees of Leland Stanford Junior University. All rights reserved.
NCBO Website Release Notes Terms of Use Privacy Policy How to Cite

Figure 1.4: BioPortal RxNORM

1.2 High Profile Applications

This section introduces three of the most futuristic and aggressive applications of knowledge-based systems: EarthCube, Gene Ontology Consortium, and NASA - NExIOM.

1.2.1 EarthCube

About EarthCube at `http://earthcube.org/info/about` describes this massive effort as:

> *Spend more time doing science and less time searching for and manipulating data.*

> EarthCube is community-led cyberinfrastructure that will allow for unprecedented data sharing across the geosciences. Its aim is to develop a framework over the next decade to assist researchers in understanding and predicting the Earth system from the Sun to the center of the Earth. EarthCube will:
>
> - Transform the conduct of data-enabled geosciences research
> - Create effective community-driven cyberinfrastructure
> - Allow global data discovery and knowledge management
> - Achieve Interoperability and data integration across disciplines
> - Build on and leverage exiting science and cyberinfrastructure
>
> This project is significant and will result in the development of a large number of complex ontologies and the capturing of big data. It should also result in new understandings of the world we live in. This hopefully will allow us to be better stewards and also help predict events and avoid major catastrophes.

1.2.2 Gene Ontology Consortium

The Gene Ontology Consortium is described at `http://geneontology.org` as:

> Gene Ontology Consortium The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products across databases. Founded in 1998, the project began as a collaboration between three model organism databases, FlyBase (Drosophila), the Saccharomyces Genome Database (SGD) and the Mouse Genome Database (MGD). The GO Consortium (GOC) has since grown to incorporate many databases, including several of the world's major repositories for plant, animal, and microbial genomes. The GO Contributors page lists all member organizations.
>
> The GO project has developed three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. There are three separate aspects to this effort: first, the development and maintenance of the ontologies themselves; second, the annotation of gene products, which entails making associations between the ontologies and the genes and gene products in the collaborating databases; and third, the development of tools that facilitate the creation, maintenance and use of ontologies.
>
> The use of GO terms by collaborating databases facilitates uniform queries across all of them. Controlled vocabularies are structured so they can be queried at different levels; for example, users may query GO to find all gene products in the mouse genome that are involved in signal transduction, or zoom in on all receptor tyrosine kinases that have been annotated. This structure also allows annotators

> to assign properties to genes or gene products at different levels, depending on the depth of knowledge about that entity.
>
> Shared vocabularies are an important step towards unifying biological databases, but additional work is still necessary as knowledge changes, updates lag behind, and individual curators evaluate data differently. The GO aims to serve as a platform where curators can agree on stating how and why a specific term is used, and how to consistently apply it, for example, to establish relationships between gene products.

The future of medical science appears to be based upon our gaining a better understanding of our own genome. So far, studies are beginning to find connections between diseases and cancers with particular genes. Preventative methods and cures are developing as a result of this understanding.

1.2.3 NASA - NExIOM

National Aeronautics and Space Administration (NASA) has constructed many ontologies in different engineering domains that are applied in space exploration. Recognizing a need to pull together all of the ontologies developed and the data collected across these domains, NASA has started the NExIOM project.

This project is described in a report entitled *"Large-Scale Knowledge Sharing for NASA Exploration Systems"* by Sidney C. Bailin, Ralph Hodgson, and Paul J. Keller. In this report, the following conclusion is given on the purpose of the project:

> NASAs goals for the next generation of manned exploration systems are ambitious, and effective knowledge reuse will be a key component of meeting the challenges. The NExIOM project represents an ontology-based approach to knowledge reuse, pushing the limits of current technology such as OWL to formalize knowledge, while employing widely accepted technologies such as XML to disseminate and collect the knowledge.
>
> The effort is in a relatively early stage, but we have already created a comprehensive set of ontologies, generated several controlled vocabularies from them, and disseminated the resulting XML Schemas to mission projects. We have begun to receive input from those projects with which to populate the ontologies with instance data. Thus, the ongoing process of knowledge evolution and ontology maintenance begins.

NASA continues to lead the world in providing the information needed to understand our universe. These efforts in knowledge-based systems will continue to lay the foundation for exploration into space.

1.3 Current Knowledge Sharing

Although sharing of knowledge is the purpose behind each of the repositories and implementations, each operates within its own set of domains. Sharing across the domains is limited by the technologies utilized and the structure of the ontologies.

1.3.1 Limited Sharing of Technology

In each of the examples of repositories and high profile applications there is little sharing of technology.

Heterogeneous architectures Each project has its own architects and technical staff that gather requirements and construct the solution to meet the needs of the specific users. These architectures are not interchangeable. They are heterogeneous and apply only to the identified usage of the ontologies. This restricts sharing.

Expensive to build Storing, organizing, and providing access to large complex ontologies is no small effort. It requires a large dedicated staff of highly-trained professionals. Only large organizations can justify this expense. Building large applications always requires a significant investment.

Lack of integration Combining knowledge from one domain application with the knowledge of another domain application, requires an understanding of both domains and their applications. Even though most applications provide methods to extract information, the integration of information from multiple sources defeats sharing. The intent of sharing is to share the formalization of logic not just the data stored in a repository.

1.3.2 Limited Sharing of Knowledge

As indicated in the previous section, sharing of data is not equivalent to sharing knowledge. This limitation on sharing occurs due to the design of the applications constructed to share the information.

Recorded to operate within a specific architecture For an architecture to provide value, each ontology must be defined according to the governance rules set forth in the design of the application. In other words, if the ontology does not meet the governance rules, it will not provide the value desired. Unfortunately, each application has its own governance rules, such that an ontology may be perfectly valid in one application and not in another.

Defined for a specific usage Just as an ontology must meet the governance rules to be useful within an application, it must also be defined with a specific usage intent. Even if the ontology may have multiple potential usages, its construction will only address the purpose of the application that will manage the ontology. For sharing, an ontology should not be restricted to a specific usage.

Mapping required Applications often provide some form of mapping of data from an ontology to another format. Other formats give the user of the data the ability to input the data into another application. Mapping requires the user to know how to operate both applications and have an understanding of the data provided and how it will be used.

1.4 Conclusion

The current state is:

- Thousands of people attend conferences each year to share ideas on how knowledge can be captured and shared. This is a strong indication of the direction of complex systems.
- The current standards developed give a provable mathematical base to information captured.
- There are multiple large repositories and applications that freely share information and are continuing to be expanded.
- Knowledge sharing is within specific application domains.
- Sharing across application domains remains a limitation to sharing knowledge.

This conclusion is a heads-up for all those that only focus on reaching a single destination. Join the crowd, look up, and recognize that KBS is changing our society. It is the best approach we have for building very large complex systems on which our society depends. But, there is a big problem with today's implementations. Our society wants fully integrated information, yet it is difficult to integrate the knowledge from multiple domain applications. The solution will be addressed in the next chapter.