Enterprise Architecture Special Interest Group (EA-SIG) Discovery Working Group Discovery Within the GIG Enterprise Services And Including a Discussion of Enterprise Architecture Patterns

Version 1.0

February 20, 2004

Prepared for the:

Defense Information Systems Agency
NCES Program Office

By the

Open GIS Consortium, Inc. (OGC)
Enterprise Architecture Special Interest Group

Contributors

Jeff Harrison – BAE – Co-Chair

A.J. Maren – EagleForce – Co-Chair

Jeff Stohlman – IBM

Mike Meyer – SAIC

Glenn Pruitt – FGM

John Clink – GD

Hans Polzer – LMCO

Mark Schiffner – FGM

Brad Mediary – Booz-Allen

Mark Young – GD

Chuck Heazel - LMCO

Change Log

Date	Author	Description	Version	Affected Pages
12/08/03	A.J. Maren	First cut at making a consistent template for White Paper, using Discovery and Mediation as guides.	0.3	All, more or less
12/12/03	A.J. Maren	Put document into format consistent with template for all EA-SIG White Papers	0.4	All, more or less
12/15/03	M. E.Young	Editing changes, comments	0.4b	All, more or less
12/16/03	A.J. Maren	Incorporating M.E. Young's edits, combine w/ other 0.4 version. Structure still pretty messy; continuing work on it.	0.5	All, more or less
12/17/03	A.J. Maren	Sorting out headings in Sections 3 & 4, still needs some attention but in better shape	0.6	middle
12/26/03	A.J. Maren	1) Outlined Executive Summary, identified needed exhibits, introduced one figure. 2) Cleaned up exhibit references. Cleaned out a lot of detailed material in Section 4; moved to various Appendices. 3) Established formatting for Appendices and subsections. Tightened up material.	0.7	Begin'g Middle End
01/02/04	A.J. Maren	Added GCSS-AF architecture and description of underlying representation levels	0.8	Sect. 3 – ***"Bkgrnd"***
01/28/04	C. Heazel	Edited and added content to achieve the following: 1) Improve the flow of the document 2) Make explicit many concepts implied in the previous version 3) Provide more information on directed discovery.	0.9	All

Date	Author	Description	Version	Affected Pages
		4) Extract and discuss general concepts that were presented in the context of specific implementations 5) Moved implementation specific text to appendix where they are presented as examples 6) Added specific recommendations		
2/03/04	A.J. Maren	Grammatical and minor editorial proofing, added one figure to show "scaling" within general inquiry space, identified two areas (highlighted in yellow) for discussion in next EA-SIG Working Grp Mtg	1.0	Several
2/11/04	C. Heazel	Expanded recommendations related to focused discovery. Expanded discussion of binding models	1.0	Section 5.1, Section 2.2
2/16/04	A.J. Maren	Careful editing pass of near-final draft; turning track changes off and accepting changes.	1.1	Through -out
2/18/04	H. Polzer	Some edits to clarify scope and user definitions	1.1.a	Several
2/20/04	EA-SIG	Official version 1.0	1.0	

Table of Contents

1 Scope

Figure 1 depicts the broad scope of GIG Enterprise Services (GES). As the enterprise services component of the Global Information Grid, GES is the infrastructure on which DoD computer applications (e.g., C2, Combat Support, Medical) rely. GES in turn relies on the GIG transport services such as the Defense Information System Network (DISN) and tactical communications systems. DISN and tactical communications systems consist of transmission systems, distribution/switching systems, Video Teleconferencing (VTC) systems, packet switching systems and other support infrastructures.

While GES relies upon the GIG transport services for the exchange between the Core Enterprise Services (CESs) and the Community of Interest (CoI) capabilities, transport is not an inherent component of GES. There are nine CES – Application, User Assistance, Storage, Messaging, IA/Security, Discovery, Collaboration, Mediation and Enterprise Service Management (ESM) services. These core services will provide a common IT infrastructure to provide reliable, secure and efficient information delivery to decision makers and the war-fighter

This document focuses on the goals, objectives, capabilities and recommendation for the Discovery Core Enterprise Service for DoD. A key future challenge is interoperation of this Service with coalition forces, Civil and Non Government Organizations (NGOs), and commercial sectors.

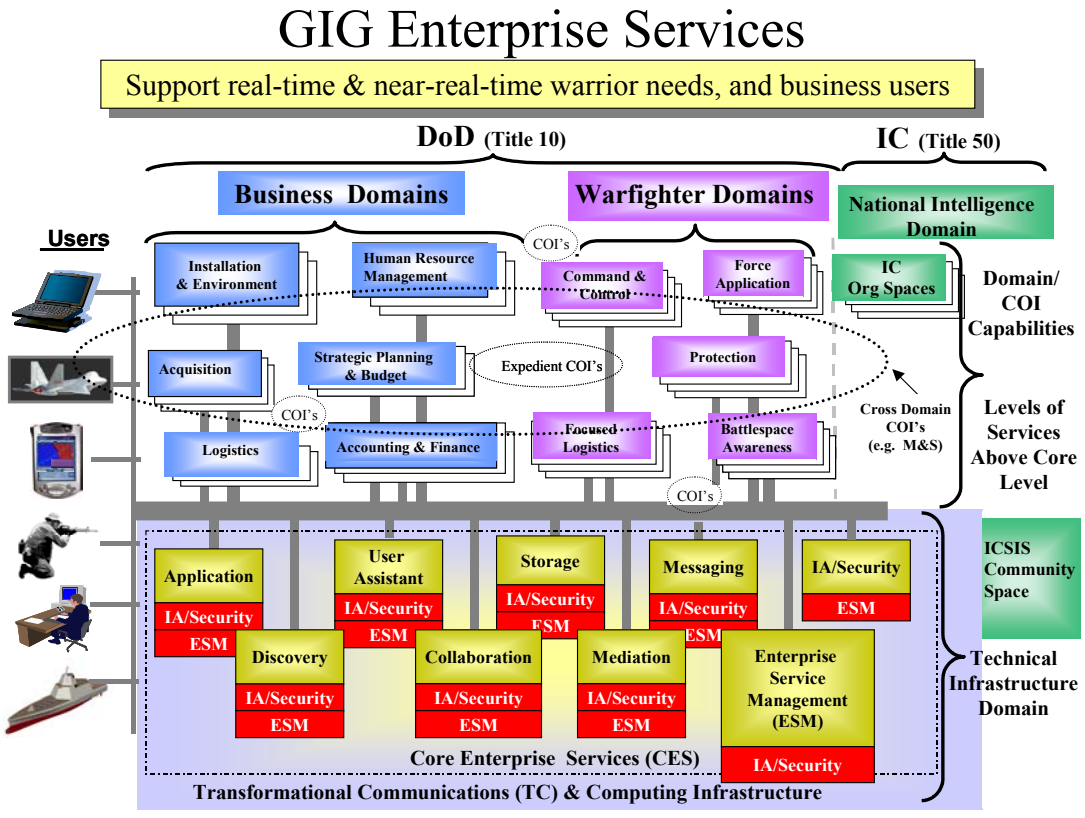

Figure 1 - GIG Enterprise Services

2 Role of Discovery in the Net-Centric Enterprise

The network centric enterprise is an environment with an almost infinite variety of resources. In this rich environment, suitable resources can be found to support almost any operational need. The problem, however, is finding the appropriate resources when they are needed. Discovery services address this problem.

2.1 Seven Step Model for Service Invocation

To understand where discovery fits into the enterprise, an understanding of how resources are accessed is in order. The seven step model for service invocation, Table 1, captures the typical process.

The Seven Steps of Service Invocation		
Step	**Action**	**Possible Instantiations**
1. Model	Develop information models (data and metadata models, process models, capability models) software/service models	*UML, Enterprise/Business process modeling tools, and other not-yet standardized implementations.*
2. Instantiate	Develop descriptive information, known as *discovery metadata,* for the models for each asset	EbRIM, Dublin Core, WSDL, XML Schema
3. Publish	Provide discovery metadata to the Discovery service	Push to the discovery service, Pull by discovery service
4. Discover (Find)	Build a discovery service request, a "query", and obtain the instances that satisfy the query – multiple instantiation methods possible	SQL, Natural Language, XML Query
5. Evaluate	Evaluate query answer effectiveness, select resource to access	
6. Bind	Establish a connection between the selected resource and the query originator	Directly or through a service broker
7. Use	Use data/service directly, or transform through data mediation service(s)	

Table 1- The Seven Steps of Service Invocation

1) Model – While discovery can occur without an a priori information model (e.g., a Google™ search on the Internet), discovery services on the GIG are likely to be much more effective if performed using established and accessible information models for information and services published or made accessible on the GIG. . Ideally every resource on the GIG will have some concept of what it provides and how it provides it. In addition, enterprise wide The creation and maintenance of information models for these shared concepts and embodying them on the GIG as an integral part of discovery services is critical for useful shared concepts for how resources are described and accessed will improve the chances that information seekers and service requestors will discover that which is sought. . Such information model-driven services will complement the more ad hoc search services such as Google that are typically associated with searches on the Internet.

2) Instantiate – Once models have been created, it is necessary to develop representations of these models that are accessible/executable by discovery services. These representations of information models are commonly referred to as metadata. At a minimum, metadata should be available to describe all service interfaces, the information context for all services, and the information model for managing that metadata. As with the models, creation and maintenance of this metadata is critical for successful discovery. While some information resources may be published on the network without an explicit service interface (e.g., a web page, although one could argue that an http request to a URL is a service interface) , eventually most resources of interest to GIG users will be accessible via a published service interface.

3) Publish – It pays to advertise. Creating a model and metadata does not make a resource discoverable. Users or service requestors (who may also be service providers) go to discovery services to find resources. For them to find a particular resource, however, the discovery service must be informed that the resource exists and how to represent it in one or more information model-based directories or service registries. Publication is the process of registering a resource with discovery services. There are a number of ways in which publication can take place including:

1) Push – the resource explicitly loads its metadata into discovery services on an unsolicited basis, presumably through a service interface that supports such a push or "posting".
2) Pull – the resource registers a URL with discovery services which then harvest the metadata through a "capabilities" interface at the resource on some scheduled basis or on some trigger event
3) Agent based – software agents, such as web crawlers, gather the metadata as they traverse the enterprise. This requires that the resource provide the metadata in a location and format that the agents can access. One class of agent performs a kind of "pull" on behalf of third parties typically representing specific interest groups or domain information brokers interested in specific types of information or services.

4) Discover – Users and service requestors who are in need of a resource will go to a discovery service to find it. This step encompasses the process of matching up user needs with published resources and returning that information to the user or software entity. A user of a service can be an end user (typically from a web portal interface), an application executing on behalf of a user (e.g., a PC client application or "servlet" on a server), or an application service provider executing on behalf of some organizational/mission entity (e.g., a "track manager" or data aggregator/integrator). A majority of this paper will deal with the capabilities required and implementation patterns of this step.

5) Evaluate – Discovery services are not perfect. Once a result has been provided to the requestor, it is necessary to evaluate that result to determine if it is sufficient or if additional discovery is required. It is not unusual for the initial result to describe more resources than desired. Multiple discover/evaluate cycles can be expected with more refined queries in each cycle, especially if the requestor is an end user (person)

6) Bind – Once a suitable resource has been found, it is necessary to establish a relationship with that resource. At a minimum this requires selecting client software that is compatible with accessing/displaying the resource or invoking the resource service interface, and providing it with the information necessary to establish a connection with the resource.

7) Use – The final step. At this point a suitable resource has been identified and an association established with that resource. All that remains is to exercise that resource's service interface to perform that task that it was needed for in the first place.

2.2 Resources as Services

Discovery services can be used to locate any type of enterprise resource. Ultimately, no matter what the resource, it will be accessed through a network protocol. Use of the resource will be enabled by software on the user's system communicating with software on the resources' system. This software-to-software interaction is a service invocation. Therefore, at the most basic level, all discoveries are service discoveries, although as discussed earlier, some services are very primitive/basic, such as accessing a web page at a specific URL. Resource discovery is the application of constraints to the selection of appropriate services, such as entering search criteria into a search engine or specifying service categories and performance parameters into a UDDI service registry request.

The concept that all resources are services has particular implications to the Evaluate and Bind steps. It is not sufficient for evaluation to assess the suitability of a resource just by its characteristics. The evaluation process must also assess whether or not there is suitable software on the users system to access the hosted service. Only if such software is available can the discovery process move forward. Likewise, the bind operation must have access to sufficient information to invoke the hostedservice. If this information is not provided as part or the resource metadata, or if the client does not have access to that service, then the resource cannot be accessed.

The need for client-side software that knows how to request the services needed to access a resource has additional implications for discovery and the enterprise framework. There are several ways to link the discovery and binding processes. Run-time binding is the case where a user already has on their machine the necessary software to access the hosting service. Currently the Web is largely a run-time binding environment with web browsers capable of accessing most resources through web servers. More complex protocols such as SOAP have evolved recently. SOAP, a lightweight protocol for exchanging structured information in a decentralized, distributed environment, facilitates information exchange between programs. With the advent of service-based architectures and the development of more complex web protocols such as SOAP, the web browser will support user access to web pages and web-based application programs, and protocols such SOAP will normally be used between programs. For example, a complete supply chain management program built upon web services, which utilize SOAP, is accessible from a browser, but the program obtains the data it presents to the browser user via SOAP from other programs, potentially executing on systems managed by diverse organizations that are part of the supply chain.

The need to support additional services leads to several additional binding models, including:

1. ***Build-time binding***: Under a build-time model, the user is the software/service developer and discovers the services required to implement the desired functionality/capability. The developer then makes the necessary modifications to the client to use thediscovered and selected services. This implies that users (i.e., developers) have the tools and authority to modify their applications and that a significant delay between service discovery and invocation is acceptable.

That is not usually the case for end users in the GES environment, although one could envision certain "superuser" tools (applications) that would permit such build time service binding by authorized end users. For example, setting up a Joint Task Force to support a particular operation might involve creating business processes and associated work flow rules, user workspaces and data repositories related to the operation, and service definitions for posting and accessing data in those workspaces. One could argue that this is a "configuration-time" binding capability, as opposed to build-time or run-time, but with the advent of interpretive execution systems, the distinction may be somewhat arbitrary.

2. ***Run-time multistep***: The necessary software is loaded on the requestor's machine (possibly requiring new levels of license management), and activated to bind with the resource as if it had always been there. This approach has numerous challengess in the GES environment including potential violation of the DoD mobile code policy, violation of many DoD Configuration Management policies for client systems, and potential violation of the Accreditation of the client system.

3. ***Proxy-brokerage***: A broker could "proxy" for the software, locate it on another machine, and direct the output back to the client machine. This approach addresses the shortfalls of the first two models. It does raise the question that if the client has sufficient capabilities to invoke the interfaces on the proxy, then why not just implement those interfaces on the service in the first place?

4. ***Service taxonomy:*** This final approach is to establish a taxonomy of well-known service types and the associated interface definitions. Client software can be written to these standard interfaces with the assurance that they will be able to perform run-time binding to services implementing those interfaces. With careful governance, client software implementing a relatively small number of interfaces would be able to invoke most of the services available on the GIG. This is a hybrid of build-time and run-time binding in which the binding is to a type of service at build time and the specifics of the service request are generated at run time. This seems to be the most probable approach for most GIG uses of discovery services. If the service interface specifications to service types include explicit versioning, context parameters, and sub-typing capabilities, services will be able to evolve and support multiple versions/sub-types of a given service on the network simultaneously. This will allow service requestors to continue working with older versions of a service type until all service requestors on the GIG have been transitioned to use newer versions of the service (at build time).

2.3 Implications

This analysis of discovery has implications on the overall architecture framework:

- Discovery metadata must include information on both the resource being discovered and the service that provides that resource,
- There should be a taxonomy of service definitions to help discover and evaluate the accessibility of resource offerings,
- There should be a limited set of standard service interfaces so as to minimize the need for build-time binding,
- There will not be a single GES resource metadata model. This implies that:

- Mediation services for discovery metadata are a critical enabler of GES wide discovery, and
- Mediation services that can integrate, federate, and orchestrate dissimilar discovery services will be required,

➢ Discovery services will be federated, including federation of repositories as well as "services" that can operate on the content of these repositories. This implies that:

- An approach to service federation that does not compromise the security policy of the federated domains must be developed, and
- A means of passing identities and credentials between security domains must be developed.

3 Discovery Background

There are many different types of discovery that must be supported by NCES services. They differ in the type of resource to be discovered, the specificity of the query, and the knowledge context in which discovery takes place. An examination of the ramifications of these parameters is detailed in this section.

3.1 Knowledge Context

The discovery process deals with finding what we know. This specifically addresses a subset of "knowability" as expressed by Defense Secretary Donald Rumsfeld.[1] Secretary Rumsfeld told a news briefing: "Reports that say something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know." He went on to say, "We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don't know we don't know." Table 2 illustrates Secretary Rumsfeld's observation as a four quadrant graph.

In the case of "known knowns," we are often performing a ***specific*** or ***focused discovery process*** initiated through a specific query. We have a strong expectation that a certain kind of information is already isolated and easily found, which allows generation of a specific answer. In many enterprise architectures, this type of discovery process is implemented using approaches very close to keyed retrieval.

"Knowability," per Defense Secretary Rumsfeld		
Knowledge	**We Know Our Knowledge ("known knowability")**	**We Don't Know Our Knowledge ("unknown knowability")**

[1] Rumsfeld , D. An ontology of knowability, *Department of Defense news briefing, Feb. 12, 2002, http://slate.msn.com/id/2081042*

Known	**"known knowns" (*)** ***Location* discovery** Starting with a known element, "locating" the *specific* information associated with that element	**"unknown knowns" (*Implicitly defined)*** ***Knowledge* discovery:** Starting with a known initial element of an event, find *more information* about that event.
Unknown	**"known unknowns" (**)** ***Knowledge* discovery:** Finding additional information related to what we know	**"unknown unknowns" (***)** ***Knowledge* discovery:** Finding new information, usually through correlations between known elements or facts

Table 2 An Ontology of Knowability

In the case of "unknown knowns" and "known unknowns," we are often performing a ***general*** or ***broad discovery process***, whereby we use a general query to retrieve information about a subject by extracting it from large corpora, and consolidating and analyzing that information into (hopefully) knowledge. This is where the "knowledge" so extracted will likely pass through several forms of representation. Different metrics are required to evaluate efficacy with each different representation and processing mechanism. As the capabilities supporting broad discovery mature within the greater enterprise architecture, we expect that the amount of "known knowns" increases because things become less unknown.

The discovery of "unknown unknowns" is beyond the scope of a discovery service. "Unknown unknowns" can only be addressed through information collection activities. As new, previously unknown information is ingested into a system, that information transitions into one or more of the discoverable categories.

3.2 Query Specificity

Discovery Services operating within the GIG ES will provide users and their agents with the means to access needed and relevant information and capabilities. Whether driven as a singular or persistent process, all discovery acts begin with a single instigation: a service request containing the query specification for what is to be discovered. (See Figure 2.)

There are two basic kinds of queries: ***specific*** and ***general***. Typically, specific queries correspond to focused discovery with specific retrieval and general queries correspond to broad discovery with general and/or specific retrieval.

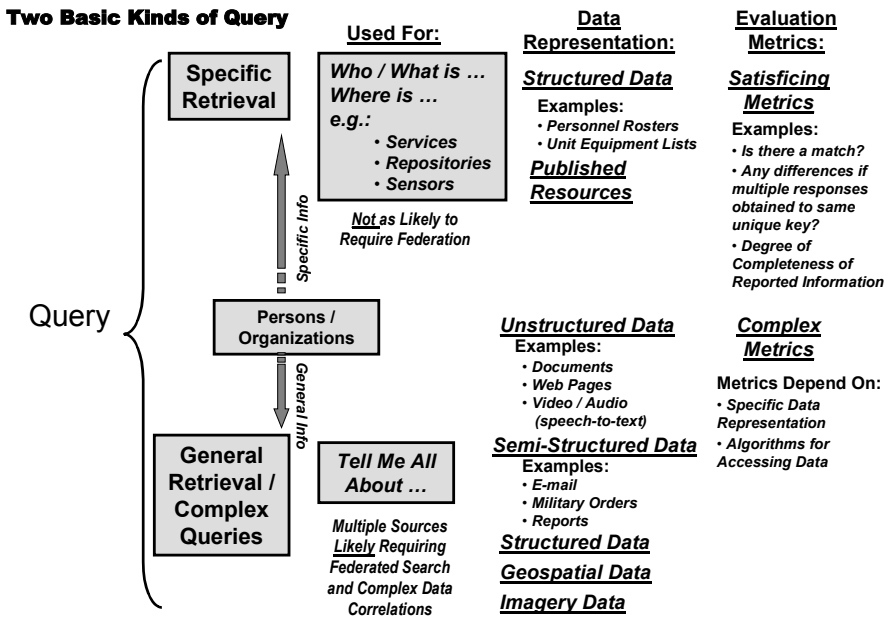

Figure 2 - Specific (Focused) and General / Complex Queries

3.2.1 Specific Query

Users and their agents typically use a specific query when they require and expect a specific answer. The discovery process supporting a specific query is typically very focused, and may be categorized as focused discovery. An example of such a query is: finding the posting for a given serviceperson, or identifying the Table of Equipment for a given dynamically-composed Marine Expeditionary Unit (MEU). Focused discovery can typically be accomplished by accessing the correct, and often singular, data repository. Further, the targets of focused discovery are typically stored as structured data. Thus, it is relatively straightforward to not only access the answer, but to perform a metric on the answer(s). For example, completeness metrics (e.g., is the serviceperson's posting available and complete?) will apply. Consistency metrics will also apply. (E.g., if multiple answers are obtained, how consistent are the answers with each other?)

Focused discovery also extends to specific queries aimed not to target so much a singular piece of information, but rather to find all instances of elements that meet certain criteria. (E.g., find all airborne IR/EO sensors currently observing a certain terrain.) These specific queries again will be subject to focused discovery with rather straightforward evaluation metrics for measuring completeness.

Enterprise Architecture support for specific queries and corresponding focused discovery is typically COI unique. It typically requires that users and their agents have significant COI knowledge of their enterprise's subject mater, data availability, and system operation before they are capable of formulating the specific queries that facilitate focused discovery. This can lead to an increasing amount of interoperability issues that may surface with COI growth and multi-COI interoperability as a federation. One COI's specific query with focused discovery often becomes a general query with broad discovery when given to a different COI. Mediation plays a critical

role in addressing this issue. By bridging the syntactic and semantic differences between COIs, mediation services will promote cross-COI discovery and a migration from general to focused discovery as confidence in the services grows.

3.2.2 Profile Query

A particular type of specific query for discovery of information about resources of special significance is the profile query. It is typically used in situations where the requestor is likely to know a common identifier for a specific resource, but wants to know additional information about that resource. Information about persons that are accessible on the GIG, i.e., GIG users, would be an example of a resource type that would merit a "profile" managed by discovery services. Note that the profile typically would not contain all information available about a specific resource on the GIG, but it would contain information about that resource that is important and useful to a broad set of service requestors on the GIG. Some of the information in a profile would include pointers to other information sources about that resource (e.g., other COI databases and application services), and the names that the resource might have in those other COI contexts.

While finding people on the GIG is obviously of great interest and utility to the broad community of GES users and service requestors, a key question that remains to be addressed is what other GIG resources merit a profile at the GES directory services level. In addition, how much information and of what type should be included in the GIG profile for that resource type. Example candidates that come to mind are organizational entities such as commands and units, computing services accessible on the GIG, as well as computing nodes such as servers and databases. However, even if a resource has a profile at the GES discovery services level, other information about that resources may be kept at the COI or even individual system level. A way to smoothly hand off discovery service requests to COI-specific discovery services at appropriate junctures in the discovery process is clearly needed – but beyond the scope of this paper. For now, let's look a little closer at the issue of profile queries, using GIG users as the example resource.

In addition to finding specific information about a person using a specific discovery query, users can also perform "profile queries." This can be of two forms: First, the system will yield back a "profile" about a given individual or resource, and second, the system can provide individuals and resources that match a specific profile.

Typically, profiles operate on structured data associated with a given resource and maintained by GES discovery services through various client applications and services. In the case of finding a specific person's profile, the response should yield that person's rank (if in service), job title / posting and operational specialty, and clearance identification. A more COI-specific profile service can report additional information.

In the case of finding individuals that match a given profile, users have the opportunity to rapidly find Subject Matter Experts (SMEs) and other people-resources. For example, a user might request a profile for an expert in neurosurgery who is available to consult at 2000 Hrs Zulu time. This involves searching a structured information repository with a simple match-logic protocol.

Adding even limited advanced query capabilities to the profiling search mechanism (e.g., concept extraction, defined in Section 4.2.1) will allow users to create and/or access useful

profiles even if they use query variants that are slightly different from what would yield precise returns under match logic.

3.2.3 General Query

In contrast, general queries are more open ended (e.g., "tell me all about *X*", "what are General Smith's views on *Y*", or "find a service that can do Z"). Users and their agents typically use a general query when they don't specifically know if the information exists, don't specifically know how to obtain the information, and/or don't specifically know what information they need. The discovery process supporting a general query is typically very broad, and may be categorized as broad discovery. Broad discovery will require additional processes and evaluation metrics beyond those required by focused discovery. Most importantly, it is the general query that will require:

- An architecture including ***multiple kinds*** of both broad and focused discovery processes and capabilities,
- Greater orchestration of multiple processes and capabilities that can meet the established capability requirements,
- Processes and capabilities to compose the potentially disparate and/or conflicting results sets into a response, and
- More complex evaluation processes and capabilities, with a different evaluation metric likely to be required at each processing level.

Enterprise Architecture support for general queries and corresponding broad discovery typically strive to be as COI independent and globally accessible as possible.

Within a COI, users and their agents often use broad discovery when they do not have the sufficient level of COI specific enterprise knowledge or access; which may or may not be intentional. In addition, they may uses broad discovery when the more focused, COI dependent capabilities are not meeting their current needs. In this case, the COI independence and global accessibility provided by general query support is leveraged in an attempt to overcome unintended barriers to processes, data and/or capabilities within the COI.

When operating across COIs, users and their agents must typically start by using the available broad discovery capabilities. One reason is because they often don't have the levels of knowledge or access required across multiple COIs for a significant amount of COI unique knowledge and capability to be immediately usable and/or understandable. Typically, a user or their agent will use broad discovery to support a learning process the goal of which is to realize more specific results; and a faster process for obtaining similar results. In short, broad discovery support is primarily leveraged by (often advanced and/or talented) users and their agents to achieve results approaching those similar to focused discovery, but without the specific foreknowledge (like the name of the specific resource being sought) required to generate a specific discovery query. In most enterprises, the consistent use of general queries should be addressed and (typically) reduced by maturing the enterprise architecture throughout its life cycle.

With general discovery, the imperative goal remains to allow nearly any authorized user and their agent the ability to begin knowing and understand any COI's data, processes, and

capabilities. However, general discovery may typically be most appropriately used by a fairly small number of users and their agents. This smaller group helps determine the more specific capabilities requirements for specific enterprise architecture improvements that can be addressed within the enterprise architectures lifecycle maturity process (this process may be totally manual, totally automated, and/or any combination in-between). It's these users who are most responsible for realizing the greatest amount of value the multi-COI Enterprise Architecture is capable of providing throughout the entire enterprise lifecycle. It's not clear to me that this paragraph is really valid and appropriate, or what message it is really trying to send. It's precisely the naïve user that is most likely to need broad general discovery queries to find what is needed.

3.3 Discovered Resource Types

The types of resources that can be discovered in an enterprise are nearly infinite. Typical discoverable resources include:

- People – individuals and information relating to an individual
  - "Specific queries" on a person will yield "known" information,
  - "Profile queries" will yield either a profile of a given person or respond with persons who match a given profile, and
  - "General queries" about a person will yield a wide range of information associated with that person.
- Organizations – organizations and information relating to an organization
  - "Specific queries" on an organization will yield "known" information,
  - "Profile queries" will yield either a profile of a given organization or respond with organizations who match a given profile, and
  - "General queries" about a organization will yield a wide range of information associated with that organization.
- Services – software entities that can be invoked over the enterprise network.
- Symbols – Different operational environment use different symbology to represent information. It is desirable to allow users on the enterprise to access any data and view it using the symbology that they are accustomed to. This suggests that standardized symbol for the different operational environments could be defined and provides as an enterprise resource for discovery and access.
- Repositories – Different data collections will house different kinds of information. High-level metadata associated with the *collection as a whole* will allow a user or the user's agent to determine whether or not a given repository should be used as a possible data source. This high-level metadata will also help users and their agents to determine necessary services for accessing the data. For example, a repository containing structured data will be accessed with different services than a repository containing free text. Repository descriptor metatags will also identify the ***security credentials*** required for access to the repository.

The kinds of data that can be held in various repositories will be of different types. These can include:

- Structured information – information that is represented in a well defined, knowable structure (e.g. databases)
- Unstructured information – information that has little or no structure (e.g. free text, voice traffic that can be converted to free text, speech or text accompanying video, etc.) This information will typically either be indexed as it is entered into a repository, or indexing can be applied to it, resulting in a set of content-based metatags associated with each element of the corpus. The indices will facilitate concept-based searching.
- Semi-structured information – information that has some or a flexible structure (e.g. XML and HTML.) This would include some information sources that carry descriptive metatags with them; e.g., metadata associated by a human or machine with an image. In addition, email traffic, radio traffic, and certain documents (e.g., reports) contain some degree of structure within known and associated data fields, or within the content (e.g., spoken identifiers in radio traffic).
- Information feeds – not all information is static. Sensor, video, and audio data, for example, are often provided as real-time information. Accessing this type of resource requires the establishment of a persistent relationship with the resource so that information can be delivered as it occurs. Some live data feeds also provide associated text.
- Stored video and image data. Some of this can have associated audio tracks, or other associated sensor data (e.g., GPS data associated with a reconnaissance video sent back from an UAV). Some video and image data will have useful metadata associated with it, done manually or automatically, or as a result of a process (e.g., "change detection") applied to the feed.

In addition, there are several different high-level forms of information that can be discovered. These include:

- Schemas – schemas describe the structure of information. In any enterprise of any size there will be many information models. The ability to discover and access schemas describing those models is a critical requirement for an interoperable enterprise.
- Ontologies – where schemas capture the structure of information, ontologies capture the meaning (semantics) of information. As an enterprise grows it becomes necessary to be able to translate information both in terms of its' schema as well as its' meaning. This requires the discovery and access of representations of meaning.
- Taxonomies – identify the specific way in which a given ontology is expressed within an organizational structure. E.g., all military service branches use aircraft, but their organizational structure for defining the aircraft and their use can vary from one service to another. A given entity (a person, an aircraft, etc.) can "inherit" properties from more than one "taxonomy." For example, a person can inherit "time in grade" from one taxonomic classification and an expertise rating from another taxonomy.

These different resource types all carry implications for discovery. All require their own unique metadata. Some have implications as to how they can be accessed. They differ greatly in volatility, from long-lived schemas to time-sensitive information feeds. The challenge for discovery services is to accommodate these differences while maintaining as much commonality as possible. This also suggests that GES-level discovery services should include a set of services that allow COI's to specify new resource types and associated metadata repositories which will

be registered and accessible from GES level discovery services. In other words, NCES-level discovery services should be inherently extensible to accommodate new resource types without having to deploy a new version of GES discovery services software.

4 GES Discovery Services

Discovery capabilities can exist as discrete services or as an integral part of some other GES or COI/application service. For example, an information management service would provide discovery capabilities so that customers can locate information within that service that they need. Another example is the need for user "presence awareness" in the collaboration core service, related to the user profile managed by GES discovery services, but also supported by security and enterprise service management core services. Discussion of discovery in such contexts is not within the scope of this paper. The discovery services being discussed here are not affiliated with any particular information set or service. Discovery in the context of other GES services will be discussed in the white papers for the respective services.

4.1 Discovery Services – Specific and Profile Queries

In order to discuss discovery services, it helps to have a generic functional model for a discovery service. The model shown in Figure 3 will be used in this paper to ***describe the case where the resources have posted metadata about themselves.*** This will include both metadata describing content as well as describing a service. This model describes four components to a ***specific query*** discovery service:

- Metadata Processing – the ingestion of metadata describing a resource and any processing done to add value to that metadata. This includes any data about the resource pushed or pulled from the resource itself or from other network entities (e.g., intelligent agents)
- Query Service Processing – the receipt of a query from a customer and any processing done to add value to that query and ensure that a response is delivered to the service request or is otherwise properly dispositioned.
- Metadata Storage – the metadata collection describing all resources known to this discovery service.
- Match Logic – the processing logic that identifies the resources that meet the query criteria based on the content of the Metadata Storage.

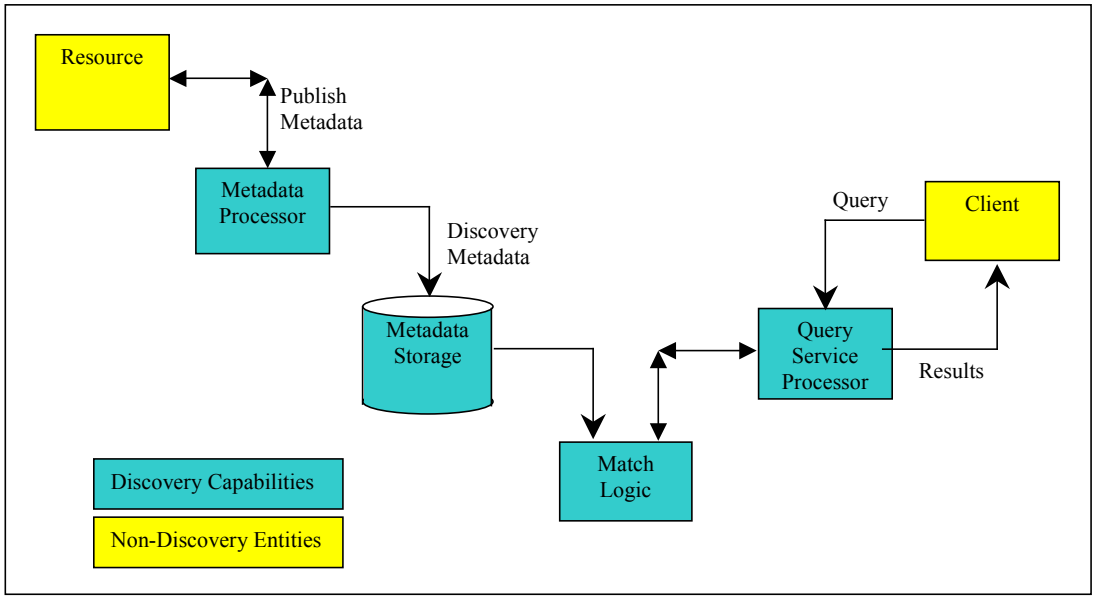

Figure 3 - Functional View of Metadata-Based Discovery

4.1.1 Match Logic-Based Discovery

Match logic is usually a simple keyword match between query parameters and metadata elements. These systems are limited to handling simple queries. They can only identify "known-knowns." The basic Google™ query is an example of a match logic-based query. Most queries against structured databases perform match logic.

4.1.2 Metadata Based Discovery

Metadata-based discovery is the most basic and most common discovery service. These systems perform little or no processing on the published metadata or on the query. Medadata-based discovery is, however, capable of discovering any resource type contained in the information model developed for GES. Figure 3 illustrates the process for metadata-based discovery

Maturity:

Most of the discovery capabilities available today commercially are Metadata based services. Examples include:

- LDAP
- UDDI
- EbXML (EbRIM)
- Z39.50

Limitations:

To be effective, the service provider and service consumer must have a common understanding of both the service invocation protocols and the metadata model of the discovery service. Over a large enterprise, this common understanding is difficult to achieve because of the inherent

diversity of perspectives and concerns across the enterprise and the dynamic nature of any large enterprise. This suggests the need for information brokers and mediators that can bridge the gap between different information models and diverse perspectives, as well as buffer changes caused by the evolution of the enterprise. Thus the GES discovery metadata model is likely to be diverse as well and will probably need to support multiple service invocation protocols.

In addition to the diversity of metadata models, the fidelity and usefulness of the discovery process is governed by the richness of the discovery metadata, in level of detail, currency/latency, and in breadth of coverage of enterprise activities/resources. More robust metadata enables greater fidelity in discovery. However, robust metadata imposes additional (and often unimplemented) requirements on resource providers and GIG infrastructure in terms of processing complexity, computing resources, and network bandwidth.

4.2 Discovery Services – General Query

In order to implement Discovery within a Net-Centric Enterprise Service (NCES) environment operating on the Global Information Grid (GIG) Enterprise Architecture (EA), we distinguish between the functional capability provided by a discovery service and the actual discovery methodology. Specifically, capabilities will encompass all aspects of Discovery as it pertains to identifying necessary processing steps and levels, referencing to the full available Enterprise Architecture, and selecting functional components from that architecture as needed.

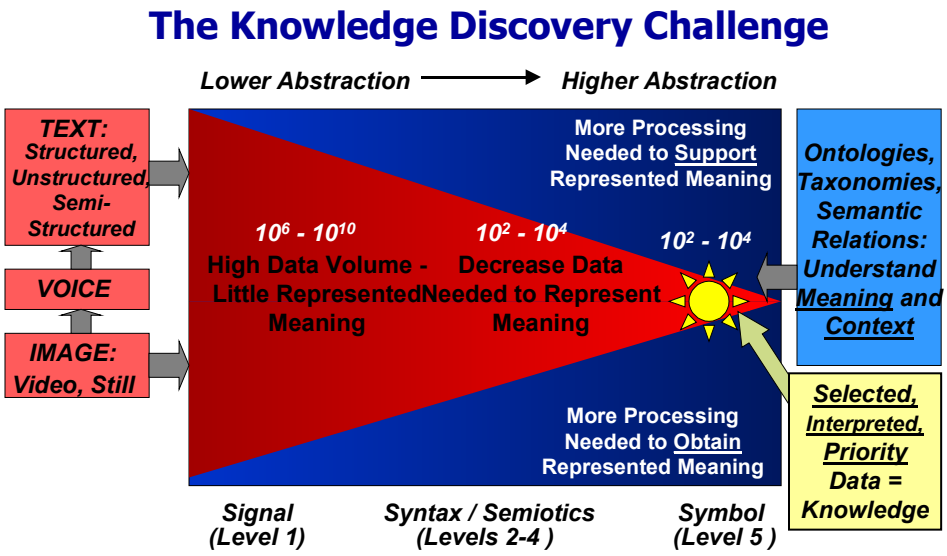

Figure 4: The Knowledge Discovery challenge is to scale down the very large-size corpora elements that are passed to the more computationally intensive but significant capabilities.

The ***knowledge discovery challenge for general discovery*** is to scale down the very large-size corpora elements that are passed to the more computationally intensive but significant capabilities. This is the requirement that will most drive the selection, federation, and orchestration of different KD services. Certain services that offer high value come at a very high computational price. In order to successfully handle large corpora, there must be a means of extracting elements that have a high likelihood of containing valuable information. These are the

elements that should be passed to the more computationally complex capabilities for further processing, as was shown in ***Figure 4***.

One way in which this can be done is to define different kinds of general discovery tasks, based on the kinds of data that are being processed and the algorithms that perform the processes. This yields a mathematical basis for describing different "levels" of general knowledge discovery. (Appendix B elaborates the "Levels" approach.)

4.2.1 Concept Extraction

Concept Extraction is typically the first step in general knowledge discovery. It extends the metadata model by identifying and incorporating concepts and concept-based descriptive metatags that are not explicitly in the published metadata. This capability is usually applied at the metadata processing stage, thereby providing a richer set of metadata against which to evaluate a query.

Maturity:

Some work has been done in this area particularly in the area of geographic locations. Geocoding software is capable of extracting location related information, such as place names and addresses, and establishing the geographic location (lat, long) associated with that information.

Limitations:

Concept extraction is limited to specific sets of concepts and the limited number of algorithms available to perform concept extraction. Concept extraction in the general case is a very difficult problem and it is not clear that discovery services at the GES level is the most appropriate way to address this capability/need. Concept extraction becomes more tractable in situations where domain context is well known and constrained. This suggests that most concept extraction on the GIG would be done at the COI level or at the specific mission application level. A key issue here is to what extent such domain-specific services would publish information to the GES level discovery services. It seems probably that a collection of information brokers will evolve over time that digest information published by domain/COI-specific resources and package it into manageable form for use by GES level discovery services.

4.2.2 Concept Correlation

Concept Correlation builds on Concept Extraction by establishing associations between related concepts. For example, a user requesting information on XML may also be interested in Web Services, parsers, and HTML.

Maturity:

Much work has been done in this area particularly in the area on-line retail, as well as existing COTS systems performing correlation after concept extraction has been done. For the GES, this capability needs to be automated. Research and development in this area will be required.

Limitations:

The same limitations discussed under 4.2.1 apply here. In addition, concept correlation is a very domain knowledge-intensive process. While some automation in limited domains has been achieved, today concept correlation tends to be a manual process. Automated processes are

computationally expensive (order of N^2); this is why concept extraction is typically used as a front-end process. In addition, concept association links can first wander extensively and second be too hard to extract as specific association sets when the corpora containing the concepts becomes too large and diverse. This is another reason why the inputs to a concept correlation tool should be the results of previous "extractions."

4.2.3 Syntactic Discovery

Syntactic Discovery introduces natural language to the query processing. Query processing identifies the "relationships" (verbs) linking "concepts" (nouns) in a query. This yields an "intelligence primitive" that is assessed against the discovery service holdings. In some cases the publish metadata may undergo similar processing allowing for better syntactic matching between query and resource.

Maturity:

Some work has been done on natural language discovery *Ask Jeeves* is an early and primitive example; more recent COTS systems have been developed. This is also an area in which capabilities are being rapidly developed and released.

Limitations:

See the discussion under 4.2.1 and 4.2.2. Automated processes are computationally intensive (order of $>N^2$).

4.2.4 Context-Based Discovery

Context Based discovery recognizes that the meaning of a term depends on the context in which it is used. By analyzing both the publish metadata and the query within their operational context, additional conceptual information can be extracted to support the discovery process. A key requirement for this approach to work is that resources need to be context-aware, and that service providers include operational context parameters as part of their service interface definitions on the GIG. This matches up with the discussion of diverse perspectives across DoD and the need to support multiple information models in the metadata, typically along operational context lines.

At this level, it is possible to extract "information primitives." This can also include people, places, and things that are recognized as such. It can also identify geo-specific places, such as Paris, France (and distinguish the "Paris in France" from any other Paris, including Paris of Troy – which is not a location at all), and provide coordinates to a geo-specific location. This requires use of certain COTS tools that are designed specifically to provide latitude / longitude correlations given a properly "named" place.

Maturity:

There are many methods for Context Based discovery but it is not clear that any are ready for operational deployment.

There are existing COTS tools that will perform geospecific coordinates given appropriate place names as inputs. Metacarta is one such example. There are other COTS tools that can extract known places with some reasonable degree of maturity.

Limitations:

Computationally intensive. May require the discovery of user and resource operational context through information services they request or provide. Over time a taxonomy of operational context types and naming services for specific instances of some of these context types (e.g, Operation Iraqi Freedom) will need to be developed and supported on the GIG

4.2.5 Semantic Discovery

Semantic Discovery enhances the matching of requests to resources by using the "meaning" of publish and query information as well as structure and concepts. This capability is enabled through the creation of ontologies and taxonomies to capture communities of common vocabulary. Using these resources, semantic discovery processes can evaluate information within its original and target semantic context to enhance matching.

Maturity:

Emerging. This capability leverages work done in the semantic web and related research. Recently WC3 approved DARPA's Web Ontology Language (OWL) – derived from DARPA's Agent Markup Language (DAML), and Resource Description Framework (RDF), as international standards. However, it will be some time before commercial products incorporate these standards.

Limitations:

Very computationally intensive in the general case. Requires a long-term investment in representing the organizational and/or knowledge infrastructure of the enterprise through ontologies and taxonomies. However, limited vocabularies in specific domains make it feasible to begin implementing useful semantic discovery services in GES for resources of great operational significance. For example, discovering military unit capabilities and current readiness status is a service that is readily achievable on the GIG.

4.3 Discovery Methodologies

Discovery requires multiple kinds of tools, interacting with each other. The GIG ES concept is that operations will not be limited, even within a given representation level, to a single tool. Rather, federated search and discovery can take advantage of whatever tools are available. This approach enables self-healing, in that if one tool is not available for a task, other tools can be selected to perform the same or similar function. We note that the majority of tools that can be considered are already owned and operated inside certain and specific elements of DoD.

4.3.1 Single service

The simplest approach to a discovery service is to provide it as a single, monolithic service with a stateless interface. Refinement of a query is performed by enhancing the query itself. Each refined query is issued against the entire metadata set. Discovery services using Web protocols often use this model.

4.3.2 Single service with feedback

A more common approach is to provide a statefull interface to the service. This allows the user to issue queries against the result set of the previous query, enabling rapid refinement of the

result set with minimal processing by the discovery service. Relational databases implement this model.

4.3.3 Federation

To be useful, a discovery service must contain an accurate representation of the available resources. As an enterprise gets larger, maintaining the currency of a single central discovery service becomes an unmanageable task. Most large enterprises address this issue by deploying multiple discovery services within local communities of interest. This leads to the problem of how to discover resources held by a discovery service outside the local community, or how to discover resources held in a local community at the enterprise level.

Federation is the process of one discovery service forwarding a query on to other discovery services and joining the results into a single response. As a result, a large collection of discovery services can be accessed through a query directed to just one.

Federation has been supported by relational databases for some time. Current versions of the UDDI and EbXML Registry specifications support federation as well.

4.3.4 Orchestrated Discovery

In Section 4, the different discovery capabilities were discussed. Using Orchestration services, discussed in the Mediation white paper, these capabilities can be brought together into a workflow such that the individual capabilities can be invoked individually or as a single integrated service.

Orchestrated services are available today.

4.3.5 Orchestrated with Controlled Feedback

Orchestrated Discovery provides a process chain with a static flow of information and control. By adding control logic, the orchestration service can create feedback loops within the process flow and control when and how those loops are executed based on intermediate results. This allows the discovery process flow to adapt somewhat to improve the performance and accuracy of the discovery process.

4.3.6 Orchestrated with Reasoning-Based Feedback

Orchestrated Discovery provides a process chain with a static flow of information and control. By adding intelligent control logic, the orchestration service can create an optimal process flow by enabling feedback loops when and where they will provide the most value. This approach will provide the most value from a collection of discovery capabilities deployed for general query processing.

5 Recommendations

5.1 Immediate – Today

5.1.1 Deployment

Sufficient discovery technology exists today to begin building the GES discovery services. This is particularly true for the case of "known knowns," including most queries against structured data. A summary of discovery related standards is provided in Table 3. We recommend that LDAP and UDDI compliant services be deployed as the first phase of the GES. These services will support the discovery of individuals, organizations, and services. Integration of these services with existing capabilities should be accomplished where possible. Each of the DoD Services has done extensive work in developing "enterprise" level directories for their users, mostly using LDAP-compliant commercial software implementations. Leveraging this work through a federation approach is potentially a "quick win" for the NCES program by standing up a global DoD directory service as a federated directory. It may also make sense to select one of the Services' directory initiatives as a "best of breed" and adapt it to include the other Services needs and content – but this is likely to be a politically sensitive approach.

Discovery of -	Applicable Standards
Individuals (specific and profile)	LDAP, UDDI, EbXML, ICML
Individuals (general)	ICML, other semantic-based standards (e.g., OWL, DAML)
Organizations (specific and profile)	LDAP, UDDI, EbXML, ICML
Organizations (general)	ICML, other semantic-based standards (e.g., OWL, DAML)
Services	UDDI, EbXML
Security Credentials	LDAP, ICML, SAML
Services – Build time	UDDI, EbXML, Z39.50
Services – Run-time	EbXML, OGC Catalog, Z39.50
Information – Structured	SQL, OGC Catalog, Z39.50
Information – Unstructured	EbXML, web crawlers, OGC Catalog, Z39.50, ICML, OIL, DAML
Information – Semi-structured	EbXML, web crawlers, OGC Catalog, Z39.50, ICML, OIL, DAML
Information – Real-time	EbXML, ICML, OIL, DAML
Schemas	XML
Ontologies / Taxonomies	OWL

Discovery of -	Applicable Standards
Symbols	MIL STD 2525C, NTDS, NATO

Table 3 - Discovery Standards

5.1.2 Research

The initial discovery capabilities proposed are only the first step in deploying GES discovery services. Further deployments will require research into the following areas:

1. **Federation capabilities:** UDDI and EbXML Registry specifications provide for the federation of discovery services. However, these are recent additions to the specifications. Compliant products should be investigated and pilots exercised to assess the maturity of this technology. Shortfalls identified through this process should be taken back to the respective organizations and used to influence future versions of the specifications.
2. **Registry Metadata Model:** A robust GES Registry Information Model (GES-RIM) for focused discovery does not exist. Development and testing of such a model should be one of the early objectives of the GES effort. An evaluation of existing registry metadata models for both services and data has been performed by the Open GIS Consortium. This study should be reviewed for it's applicability to the GES environment.
3. **Registry Population:** Focused discovery can only be successful if the metadata store of the discovery service is populated. Approaches that automate the population of discovery metadata need to be investigated. Harvesting (pull-based publication) of discovery metadata is not a part of the existing discovery specifications. However, Discovery Services that support Harvesting do exist. In view of the potential savings in metadata maintenance, investigations into the suitability of existing harvesting COTS products as well as the potential enhancement of standards based COTS discovery products should be explored.
4. **Storage Services:** The scope of this paper has been on dedicated discovery services. The discovery problem, however, extends down to the individual data providers. Should all managed data providers (storage services) support a discovery interface? This would enable discovery to progress through increased levels of granularity as the users drill down to the data they need.
5. **Mediation Capabilities:** Mediation capabilities are a key requirement for federating discovery services across COIs. R&D resources should be allocated to mature this technology, particularly in the areas of adaptation and transformation. It would be best that the mediation capabilities be incorporated as part of the discovery service as opposed to a separate distinct service.
6. **Benchmarking:** Discovery capabilities for Service Oriented Architectures exist today. As the GES Discovery Services evolve, these existing implementations should be studied to benefit from their experience. These existing implementations include:
   - Canadian Geospatial Data Infrastructure (CGDI) - http://www.geoconnections.org/CGDI.cfm/fuseaction/home.welcome/gcs.cfm
   - National Spatial Data Infrastructure (NSDI) - http://www.fgdc.gov/index.html

- NASA EOSDIS - http://spsosun.gsfc.nasa.gov/eosinfo/Welcome/index.html

7. **Tools:** We recommend that an analysis of COTS products suitable for providing each of the capabilities identified in Section 4.2 for General Discovery be conducted. There is at least one, and typically at least 2-4, COTS tools available for each of those defined discovery capabilities. ***Appendix E*** presents a "strawman" evaluation matrix by which these existing tools can be analyzed.
8. **Orchestration Capabilities:** In addition to tools to provide well-known functions against different types of data, we should immediately identify COTS tools that can perform orchestration of either or both services (e.g., search) or repositories. Also, systems that can provide a framework for integrating two or more COTS tools so that they can "communicate with" each other, and allow feedback, need to be identified.
9. **Architectures:** It is clear from the discussion of different general discovery functional areas, that different capabilities will be needed to perform the different functions. It is not likely that any single vendor will provide all the desired functionalities within a given suite. Even if this were to be the case, the prospect of federating different functional services along with repositories requires that a "discovery-wide" architecture approach be used. We recommend that existing architectures in place for DoD be examined for both their immediate use and their potential for supporting enhanced capabilities. The particular concerns should be:
   - Ability to incorporate different tools or capabilities to provide different functions, including the ability to federate multiple tools at a given "level" of processing (e.g., for concept extraction),
   - Ability to allow access to multiple data sources and/or repositories; allowing "federation" of resources,
   - Ability to allow structured feedback from one form of processing to another, e.g., using the results from "concept correlation" to generate a more refined "concept extraction" search,
   - Ability for a system to profile an individual user or other significant resource,
   - Ability to allow users to perform profile-based queries, and to automatically construct profiles of users and other resources that can be used as the basis for answering profile-based queries,
   - Ability to readily associate context-specific information extracted about a person, place, thing, or event with appropriate context-based representation. For example, the discovery service could provide information about a user in his current geo-location with geo-referenced coordinates, and then provide basis for geo-spatial association and reasoning about that user and related entities such as his unit or operation,
   - Ability to readily extract and identify specific references to "known" persons, places, and things from unstructured data, and use these extracts as the basis for generating "specific queries" as well as more advanced pattern-finding methods on structured data,
   - Ability to allow information extracted from structured (specific) queries to be pushed against unstructured corpora as well as geospatial corpora for more general discovery, and

- Ability to provide the user with a comfortable and seamless environment for information visualization and query formation / reformation.

Appendix B provides a high-level overview of one DoD KD architecture, the GCSS-AF. We recommend that this be evaluated, along with other existing DoD architectures, and that based on this evaluation, an architecture be either selected or proposed. We also recommend that the COTS Tools and Services evaluation be used to identify and characterize an initial set of COTS capabilities. This set should provide a reasonable selection of tools ***at each level of processing*** for potential use in a federated environment.

Two particular considerations will be: Suitability of COTS capability for use in a federated GIG environment, and accessibility within an architectural framework.

Following best-of-breed identification, two important steps will be to identify which tools are already owned and in use, and can "fill in" an architecture with minimal cost, and/or those that can be readily obtained to provide necessary but currently missing capabilities within an initial architecture.

These steps can all be performed within a relatively short timeframe. Once accomplished, a subset of repositories should also be selected for initial test. The initial architecture should be rapidly prototyped to operate against the selected repositories to perform specific, profile, and general queries. Analysis of results performed against a known testbed will identify greatest needs for the next stage.

5.2 Vision: 5 - 10 Years

It is not likely that the tasks laid out for the next five years will be finished within that time period. They should continue over the longer period of time providing increasingly robust and accurate discovery capability.

A Glossary of Terms

Ad-Hoc COI – an operational COI that forms in response to immediate events. Ad-hoc COIs come into existence to address an issues and disband once that issue have been resolved.

Application Schema – An application schema provides the formal description of the data structure and content required by one or more information communities. --- set of conceptual schema for data required by one or more applications.

COI – Community of Interest.

DCP – Distributed Computing Platform

F**eature** – abstraction of a real world phenomenon or attribute of a system

Federation – an IT configuration where organizations and systems collaborate without a single management framework.

GML – Geographic Markup Language

Information Community - a collection of people (a government agency or group of agencies, a profession, a group of researchers in the same discipline, corporate partners cooperating on a project, etc.) who, at least part of the time, share a common digital geographic information language and common spatial feature definitions.

Interface – named set of operations that characterize the behavior of an entity

Jurisdiction - an administrative entity with a single management authority that can establish standard policies, procedures, and technologies. All systems within a jurisdiction are subject to this management framework.

Metadata – data about data.

OGC – Open GIS Consortium

Ontology – the working model of entities and interactions in some particular domain of knowledge or practices, such as electronic commerce or "the activity of planning." A set of concepts - such as things, events, and relations - that are specified in some way (such as specific natural language) in order to create an agreed-upon vocabulary for exchanging information In artificial intelligence (**AI**), an ontology is, according to Tom Gruber, an AI specialist at Stanford University, "the specification of conceptualizations, used to help programs and humans share knowledge." . One or more taxonomies can be developed for the ontology and taxonomies can be used as part of the ontology model.

Operation – specification of a transformation or query that an object may be called to execute. Also, a virtual enterprise established to achieve some real world goal (e.g., Operation Iraqi Freedom) – see Ad Hoc COI

Operational COI - a collection of individuals, organizations, and systems with similar business and information needs. Operational COIs operate across Jurisdictions and Federations and in fact are the primary reason for their existence. Operational COIs develop their own operating conventions addressing such issues as information models, policies, and practices.

Service – distinct part of functionality that is provided by an entity through interfaces accessible over the GIG network.

Taxonomy – the science of classification according to a pre-determined system, with the resulting catalog used to provide a conceptual framework for discussion, analysis, or information retrieval. In theory, the development of a good taxonomy takes into account the importance of separating elements of a group (taxon) into subgroups (taxa) that are mutually exclusive, unambiguous, and taken together, include all possibilities. In practice, a good taxonomy should be simple, easy to remember, and easy to use. However most real world entities and concepts can be viewed as belonging to multiple taxonomies, based on the operational context in which they are referenced. For example, a main battle tank is both a vehicle and a weapon system. It can also be a shelter, cargo, asset, target, etc. in other operational contexts and thus taxonomies.

Viewpoint – form of abstraction achieved using a selected set of architectural concepts and operational contexts with associated structuring/representation rules, in order to focus on particular concerns within a system development, acquisition, or virtual enterprise context.

B An Example: Discovery in GCSS-AF

Discovery requires multiple kinds of tools, interacting with each other. The GIG ES concept is that operations will not be limited, even within a given representation level, to a single tool. Rather, federated search and discovery can take advantage of whatever tools are available. This approach enables self-healing, in that if one tool is not available for a task, other tools can be selected to perform the same or similar function. We note that the majority of tools that can be considered are already owned and operated inside certain and specific elements of DoD.

Further, within the GIG, content management can also be federated, so that the same suite of tools used for discovery can also be used for content management. While discovery operates down to the word level, the content management process typically identifies documents and other elements, and associates them with proper categories within a taxonomy. The role of mediation is seen to facilitate both the processes of discovery and content management.

As we approach the semantic web, the GIG ES will use Level 3-type processes (where are these levels defined/described?) to identify relationships between distinctive, stand-alone elements (web services, repositories, etc.) and map these relationships. This will enable the discovery and content management processes to embrace the emerging semantic content of the web.

In GCSS-AF, Level 1 is already implemented for general discovery, and Level 5 is being developed. GCSS-AF already performs federated search, capable of using multiple search engines and multiple repositories.

Discovery of "Unknown Knowns" and "Known Unknowns" – General Discovery

Google™ is a current point of reference for many people when they undertake discovery. As a point of reference, Google™ is limited by more than its inability to access many types of managed data and the lack of consistently applied metadata. In fact, one of the most critical factors ***early*** in the general discovery process is to address the problem first identified by Google™ founders as "relevance ranking." The Google™ approach is to use a human-in-the-loop:

> *"The ranking function has many parameters like the type-weights and the type-prox-weights. Figuring out the right values for these parameters is something of a black art. In order to do this, we have a user feedback mechanism in the search engine. A trusted user may optionally evaluate all of the results that are returned."* (http://google.stanford.edu/)

It is clear, though, that for autonomous knowledge discovery, having human-in-the-loop processes at the low end is not only inefficient, but will lead to inconsistencies in terms of ranks produced by different people. An autonomous feedback system is clearly necessary.

Another challenge with "general discovery" is that a query can – and ultimately ***should*** - be able to address more than simple keyword-based retrieval. Users think about entities and/or concepts not just by themselves, but ***in relationship to*** other entities and concepts. Also, we know that context is not just valuable, but essential in pruning search. Further, as we develop organizational and functional taxonomies for the various services and their operations, we need a discovery mechanism that intrinsically "reaches" towards knowledge in the way that a human would expect and desire, given that knowledge (content) can be managed within an organizational / functional structure.

These considerations led the Air Force to adopt a GCSS discovery approach based on a multi-representation-level ***architecture***, rather than as implementation of a single point-source solution. The baseline GCSS-AF architecture is illustrated in ***Exhibit 1***.

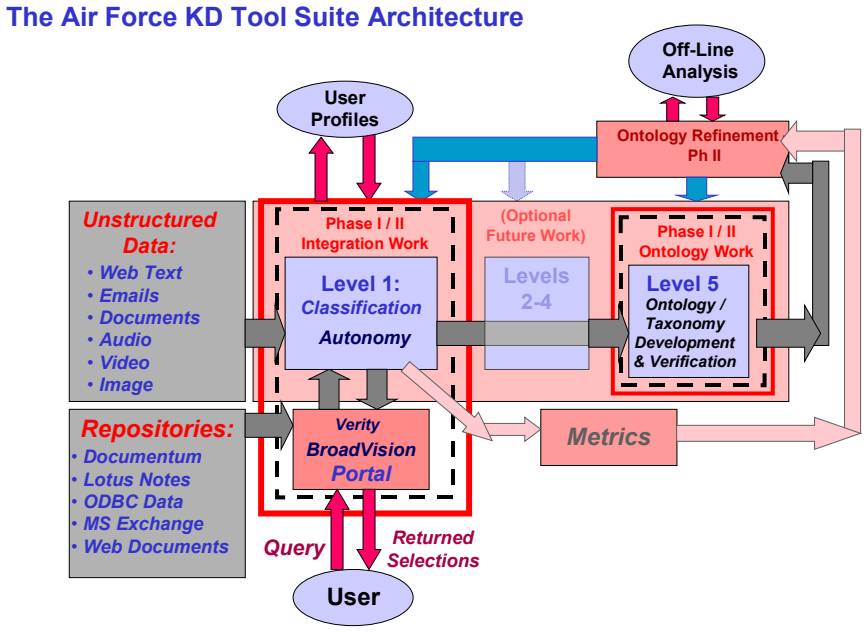

Exhibit 1: The GCSS-AF Knowledge Discovery Architecture

GCSS-AF Phase I development has been completed and the system is fielded, with an initial user base of 70,000 persons, anticipated to grow to 1.2M persons by April, 2004. Phase 2 development of "heavy indexing" is being undertaken beginning 2nd Quarter FY04.

The more complete architectural framework, on which the GCSS-AF architecture is based, is shown in ***Exhibit 2.***

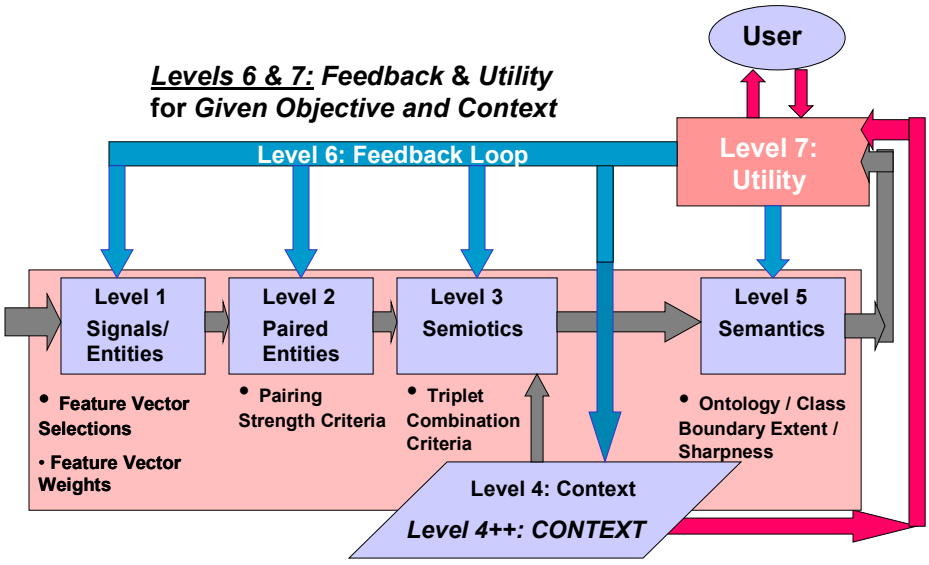

Exhibit 2: Basic five-level representation architecture for linguistic-based knowledge discovery

As a point of reference, Google™ can be viewed as approximately a Level 0.5 capability within the framework of a functional discovery architecture, for which the five representation and two control levels shown in ***Exhibit 2*** have been defined.[2]

Exhibit 2 shows the representation levels in an architectural framework for ***functional discovery***. In this sense, no single "point" solution provides the answer, although integration of multiple COTS/GOTS tools can yield desired competency.

Specific feedback loops allow for questions to be decomposed into smaller questions, and for assembly of information from multiple sources. The process of iterating intelligence-gathering and refinement allows ***non-obvious information to be gathered***. ***Information from different representations (data types) can be integrated*** within a common architecture.

While the functional elements for discovery, specified more clearly in ***Exhibit 13***, perform the actual discovery process, the way in which they are dynamically and adaptively composed to perform this function lies within the realm of federation. Specifically, functional elements must be selected, assembled, and orchestrated. The federation meta-control architecture must be able to perform self-healing of a discovery architecture, depending on the query and resources at time of query posting.

[2] EagleForce "Black Dragon" Knowledge Discovery Architecture, Patent pending. Initial concept presentation at Georgetown University Faculty Colloquium, October 2002, taught in upper-level undergraduate / graduate course at Georgetown University in Spring, 2003, and has been requested for replication at Naval Postgraduate School, as well as being briefed at NWU. Architecture was used as basis for Air Force GCSS Knowledge Discovery capability.

Discovery uses multiple levels of representation, along with control methods, as summarized in ***Exhibit 3***.

The Seven Representation Levels for Discovery		
Level	**Function**	**Methodology**
1. Concept Extraction	Identify and extract concepts (persons, organizations, geographic regions, any other conceptual entities); apply concept-based descriptive metatags to linguistic corpora elements and their segments, as well as appropriately indexed images.	*Statistically-based methods, including Bayesian Logic, enhanced with Shannon's Information Theory (and alternatively) Semantic Nets*
2. Concept Correlation	Identify those concepts that are statistically close within corpora elements	Co-occurrence matrices (N^2 process); Latent Semantic Indexing
3. Syntactic	Identify "relationships" (verbs) linking "concepts" (nouns) => yields an "intelligence primitive"	Syntactic analysis (>N^2; computationally expensive)
4. Context	1) Identify "context" associated with any "intelligence primitive" (concept-relationship-concept) 2) Enable "handover" of primitive to structured data processing and analytics 3) Enable "handover" of an event to geospatial / temporal representation and reasoning	Multiple methods, many computationally expensive
5. Semantic	Ontologies and their taxonomies, provide inputs to feedback loops governing Level 1 classification / concept categorization	Very computationally expensive; also typically long-term investment of representing organizational or knowledge infrastructure
6. Feedback Control with Utility	Control scaling and feedback from one representation level to another	Feedback loops input values to control system, modulated by utility functions
7. Reasoning-based Metacontrol	Define strategy for transitioning "knowledge" from one level to another; define strategy for feedback and "spinning off" related queries. Define strategy for identifying when alert thresholds are reached.	Business rules, schemas, rule-based reasoning, adaptive pattern recognition.

Exhibit 3: Discovery can be performed using five basic representation levels, along with two control levels.

We note that existing COTS tools are available for each of *Levels 1-5*. There are further existing capabilities that can perform the control functions at *Levels 6-7*. We further note that the architecture is illustrated for linguistic data processing. The same architectural concept extends to geospatial data and to sensor-based data. Data migration between different major representation forms (e.g., linguistic to geospatial / temporal) is not only feasible, but desired during general discovery. For example, "information primitives" extracted at linguistic *Level 3* from text-based data can be inserted as single "events" into a geospatial / temporal representation. Extracted "information primitives" also can be inserted into structured databases, or be used to generate queries into structured data.

As multiple representation levels are needed, each representation level will have its own unique metrics for ***efficiency*** and ***effectiveness***. Thus we need two general kinds of metrics; "aggregate" metrics that apply to things like the number of people that can be cataloged, etc., and functionality metrics, that can be applied to each representation level.

C Role of Representation in Discovery

Our human brains devote about one-third of their processing power to handling linguistic-based information, and about another one-third to processing visual and geospatially-based information.[3]When we ask ourselves questions, and answer them for ourselves or to others, we use both of these very fundamental forms of "knowledge representation." In fact, the biggest challenge is not just working within a single knowledge representation form, but rather knowing when and how to integrate both linguistic and geospatially / temporally-based knowledge.

This illustrates how different kinds of reasoning capabilities need to be brought into play for answering the question. The three kinds of representation needed to support this reasoning are ***linguistic, geospatial/temporal,*** and ***data-intensive***, as illustrated in Figure C-1.

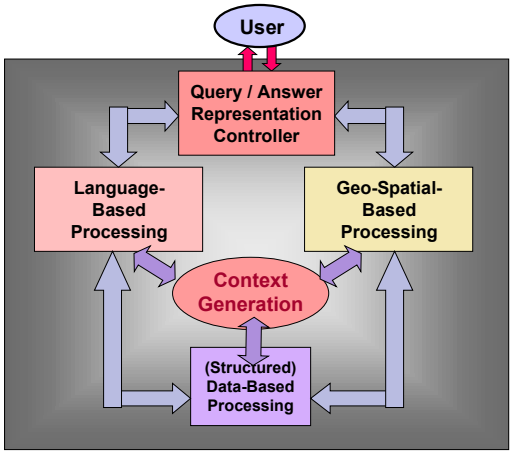

Figure C-1 - Multiple kinds of knowledge representation needed for integrated discovery

Humans also work with facts. This is an area where computational systems have excelled; the various analytics that extract knowledge and context from structured data are included within an advanced architectural framework.

An advanced discovery architecture requires that different representation forms (text, geospatial / temporal, structured, etc.) and different representation levels within each form be used and guided by architecture meta-control. ***Specific feedback loops allow for questions to be decomposed into smaller questions, and for assembly of information from multiple sources***. The process of iterating intelligence-gathering and refinement allows ***non-obvious information to be gathered***. ***Information from different representations (data types) can be integrated*** within a common architecture.

[3] Kolb & Wilshaw, *Fundamentals of Human Neuropsychology*, 3rd Ed.. (1990, Freeman); see also Volume II.

D Role of Taxonomy During Discovery

Taxonomies are a way to organize documents or web pages into logical groupings, based on their contents. Ideally, documents discussing the same subject will be grouped together into one of the taxonomy's categories. A corporate taxonomy is a way of representing the information available within the organization, In its simplest form, it is a hierarchy of categories that is used to classify documents and other information within the corporate knowledge base.

Taxonomies are often organized into "trees" to make them easier to navigate; the subject-related categories and subcategories form the "branches" of the tree. Near the "root" of the tree are very broad subject categories, such as "financial management", "logistics", "Personnel" and "Medical". As a user navigates down a particular "branch" of a tree, the subject categories get more and more specific. For example, a user navigating down a "Medical" branch might then select "Surgery".

Probably the best-known example of a taxonomy is the Yahoo Internet portal. Yahoo has logically grouped the millions of web pages they index into convenient categories and subcategories. Taxonomies are also sometimes referred to as "knowledge trees" or "topics", depending on the vendor.

Once a taxonomy tree has been created, all the documents in the system are tagged as belonging to one or more specific taxonomy categories. This process is typically referred to as "categorization", "tagging" or "profiling", again depending on the vendor. Users can then browse and search within specific categories. Figure D-1 shows an example of a three-level taxonomy.

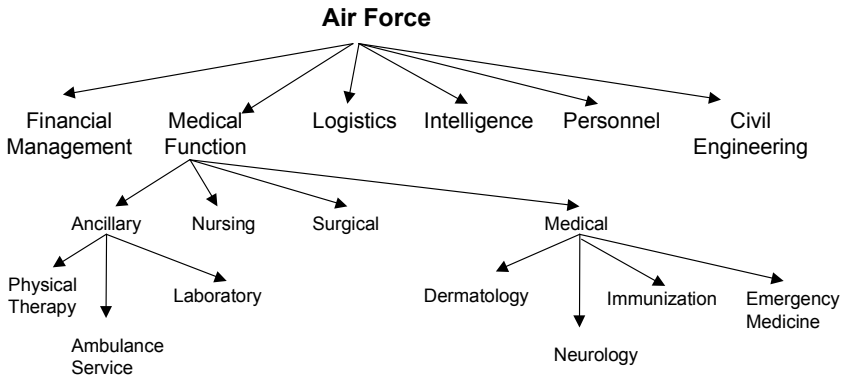

Figure D-1 - A tree-level taxonomy

D.1 The Value of Taxonomies

Taxonomies are becoming very important to organizations such as Federal Agencies as they struggle to organize their ever-increasing mountains of electronic data.

Early search engines had no problem searching through a few thousand documents that might have been stored in a single large repository; almost any search engine works fine if you only have 1,000 documents! As the number of indexed documents grew, search vendors tried again

and again to improve their search engines, some even added Artificial Intelligence to parse user queries and locate pertinent documents.

But the average user search is composed of just 1.4 words! These short one and two word queries thwarted most of those advanced algorithms. But more importantly, it has become clear that users sometimes prefer an iterative experience. They enter a one word search, and then look at the results. Based on the results, they may edit their search and try again. The early search engines were much more "one shot" oriented. Taxonomies provide a well understood structure for more modern, targeted searches.

Taxonomies provide several key benefits:

- Documents are partitioned into logical groupings which are easier to navigate
- Allows users to locate information even if they start with a single word search term
- Taxonomies facilitate iterative, drill down searches which both advanced and beginning users can quickly traverse.
- A taxonomy category can be used to limit the scope of a search, thus reducing the amount of irrelevant documents returned
- A well organized taxonomy adds "context" to documents that are returned in a search result; the category a document is listed in can convey concepts such as "relevance", "source", "authority", "public vs. private" and chronological indicators.
- Taxonomies give customer service pages and corporate portals a more professional, organized look, and an improved navigational structure.
- Taxonomies also help avoid problems with common English language peculiarities of similar sounding words, or words with multiple meanings.

For example, does the search term "sun" refer to the center of our solar system, or to the company Sun Microsystems? A user typing in this search might be presented with two branches, one labeled "Science / Astronomy / Solar System" and a second branch labeled "Business / Computer Companies / Sun Microsystems" - it would be very clear to the user which documents dealt with which concept. They could then investigate the appropriate branch further.

Or think of Ford, Ford or ford: the car, a person or a river. By looking for the other words grouped around it, classification stops you from getting a huge dump of documents just because they have the words you've searched for.

When properly implemented, taxonomies speedup employee access to critical data, dramatically increasing their productivity. Ultimately, this is the main reason companies implement the technology. Often multiple taxonomies are necessary to provide adequate user navigational assistance in searching the concept space appropriate to an enterprise. The larger and more diverse the enterprise, the more likely this will be the case. Obviously, DoD and GIG users will require a multiplicity of taxonomies to adequately represent the information resources accessible on the GIG.

D.2 Creating Taxonomies and Categorizing Documents

Once the concept of taxonomy is understood, the next question is typically "So where do these taxonomies come from?"

Different vendors have different methods for creating and maintaining taxonomy trees. Some vendors separate the creation of the trees from the process of categorizing documents, whereas other vendors combine these two processes.

There are three general type of taxonomy creation, with some vendors offering tools that span more than one type:

D.2.1 "Automatic" Taxonomy Creation and Document Categorization

Some vendors use statistical models to automatically categorize documents and arrange the subject groups into clusters that they refer to as "taxonomies." Some vendors offer this capability. In general, these "automatically" created categories bear little or no relation to the categories identified by content managers as their desired taxonomy categories. Most vendors also allow modifications or create categorization rules to have more tight control over which categories a document is place in. This is generally a time-consuming endeavor.

D.2.2 Assisted Taxonomy Creation and Document Categorization

This is the most common type of categorization and taxonomy creation. During a highly interactive and iterative process, knowledgeable personnel act as trainers who monitor the categorization of hundreds (or thousands) of documents and take actions to modify the rules the system is using. This is again very labor-intensive. Some vendors allow trainers to directly input and override key words and phrases that the system is using, while other vendors simply have the trainers indicate which documents should and should not go into each category. Trainers can also indicate that a category should be further subdivided into subcategories, which gives more precise categorization of documents.

It's important to note that some vendors offer both automatic and interactive categorization. Each vendor's product has its unique strengths and benefits, and style of interaction.

D.2.3 Professional Taxonomy Creation

Though many advances have been made in automatic or semi-automatic taxonomy creation, there is no substitute for a professionally created taxonomy. For certain applications this is still the only acceptable route. The typical motive for selecting a professional taxonomy is either the need for a very high quality tree or the desire to deploy a project quickly and avoid a lengthy setup and training period. Vendors typically offer libraries of taxonomies pertaining to specific industries such as financial, legal and medical information. They can also work with a client to create a specific taxonomy targeted to that client's exact needs.

Taxonomy, classification and search are increasingly coming together, and with good reason: they need one another.

The more complex the enterprise, the greater the need to search among multiple sources, but the one- or two-word search doesn't give much complexity in the results.

Each word can have many meanings. To solve the problem users need to narrow down the topic and the solution is to categorize. As soon as they have categorized or classified they have narrowed it down.

Combining taxonomy and classification with search gives users a map of the resources available to them. This combination of taxonomy, classification and search is becoming essential for the major search vendors. Often what users want to do is browse because they are not sure how to ask the question and the taxonomy provides a display of information that does not require the users to put the inquiry into words.

Experts now agree that Taxonomy, classification and search need one another and vendors including Autonomy, Convera, Inxight, Stratity and Verity are among those attempting to put all the pieces together.

The taxonomy is vital, but it must not be rigid. One basis for change is the need to tailor the search experience to the differing requirements of various users. The Contracting and Ops people, for example, have different ways of looking at the same information. Some things can be irrelevant to those in either group, so there may be a need for multiple taxonomies or views of the same information.

A document may be of interest to different groups of users for different reasons, and forcing it into a single predefined category may be neater but may also reduce its usefulness. Taxonomies need to be flexible, pragmatic and consistent.

By combining taxonomy, classification and search, organizations will give their users the ability to pull the precise needle from a haystack of information.

D.3 Topic Maps

"Topic Maps" is a new ISO standard for describing knowledge structures and associating them with information resources. As such they constitute an enabling technology for knowledge management. Dubbed "the GPS of the information universe", topic maps are also destined to provide powerful new ways of navigating large and interconnected corpora.

Without it, it is like a book without an index or a country without a map.

D.4 Thesaurus

A thesaurus is basically a network of interrelated terms within a particular domain. It will often contain other information such as definitions, examples of usage, etc. The key feature of a thesaurus is the relationships or associations between terms. Given a particular term, a thesaurus will indicate which other terms means the same, which terms denote a broader category of the same kind of thing (e.g., F-22 and Raptor), which denote a narrower category, and which are related in some other way.

E Strawman COTS Tool Evaluation Matrix Template

E.1 High-Level Evaluation Matrix

	High-Level Capabilities																				
	Tool Level (e.g., 0.5, 1, 1-2, etc.)																				
	Methodology*																				
	Capability Match																				
1	Multi Domain Cllctn																				
2	Data Profiling																				
3	Ent. Extrct'n/ Ctgzt'n																				
4	Data Translation																				
5	Nat Lang Recognit'n																				
6	Data Srch & Mining																				
7	Link & Temp.Anlys																				
8	Ent. Rel'nshp Anlys																				
9	Geospatial Analysis																				
10	Collaboration																				
11	Reporting																				
12	Disseminat'n																				

* Define methodologies in separate description for each COTS tool. Methodologies for general discovery will correlate directly to classification in one or more "representation levels," depending on the algorithm and the data to which the algorithm is applied.

E.2 Data Ingestion Matrix

	Data Ingestion																			
	Tool Level																			
1	**Unstructured / Semi-Struct**																			
A	Document																			
B	Web																			
C	E-Mail																			
D	Foreign Language																			
E	Audio																			
2	**Geospatial / Image**																			
A	Video																			
B	Image																			
C	Geospatially-Referenced Objects																			
D																				
E																				
3	**Structured**																			

- *= explanation should be provided with fuller details.