INSTITUTE FOR DEFENSE ANALYSES

Proof of Concept Assessment for the Use of Natural Language Processing to Maintain and Update the DoD Technologies Knowledge Base (DTKB)

Robert Rolfe, *Project Leader*

Steven P. Wartik, *Principal Investigator*

Francisco L. Loaiza-Lemos, *Solution Architect*

Anna Vasilyeva

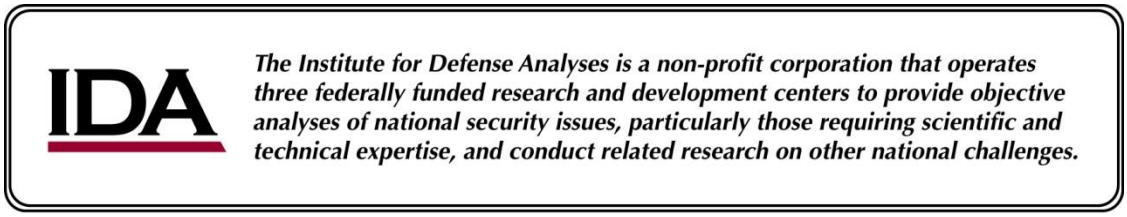
Thi U Tran

December 2015

Approved for public release; distribution is unlimited.

IDA Document
D-5685
Log: H 15-001227
Copy

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882



About This Publication

This work was conducted by the Institute for Defense Analyses (IDA) under contract HQ0034-14-D-0001, Task AK-5-3782, "Utilization of Machine Learning for Maintenance and Update of DTKB," for Director, ASD(R&E)/RD/Technology Security Office. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

Robert F. Leheny

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-5685

Proof of Concept Assessment for the Use of Natural Language Processing to Maintain and Update the DoD Technologies Knowledge Base (DTKB)

Robert Rolfe, *Project Leader*

Steven P. Wartik, *Principal Investigator*
Francisco L. Loaiza-Lemos, *Solution Architect*
Anna Vasilyeva
Thi U Tran

Background

The requirement for the Department of Defense (DoD) to create and maintain a Militarily Critical Technologies List (MCTL) was established by Public Law 99-64 on July 12, 1985. The purpose of the MCTL was to identify technologies that are critical to national security and thus require extra protections that include bans on exports and the application of anti-tamper technology.[1] The MCTL is now outdated for the majority of its intended uses. The most recent Government Accountability Office (GAO) report (GAO 13-157)[2] on the use of the MCTL notes that the value of MCTL information has significantly deteriorated, due primarily to lack of funding to carry out the necessary update activities. Sections on emerging technologies are no longer being periodically revised to reflect the current state of the art, while other sections haven't been updated since 1999.

Given the above situation, many DoD organizations now rely on alternative information sources, and the envisioned main users of the MCTL — technology export decision makers in the Departments of State and Commerce — have turned to *ad hoc* networks of subject matter experts for detailed information on military criticality when their internal subject matter experts are unable to provide this information. Other agencies are developing their own versions of a MCTL, potentially creating conflicting views of what is militarily critical, while funding efforts to produce results already obtained elsewhere within DoD.[3]

Because the requirement to review and update the MCTL *at least annually*[4] is still in force, in 2013 the sponsor asked the Institute for Defense Analyses (IDA) to assess possible courses of action to answer an inquiry into the status of the MCTL activities by the Office of the Inspector General (OIG). Among the various courses of actions (COA) explored, IDA recommended that AT&L: "re-invent, in the near term, the MCTL as a shared and integrated set of existing information sources — thereby creating a common, dynamic, classified, proprietary, DoD Technologies Knowledge Base (DTKB) reflecting

1 http://en.wikisource.org/wiki/Page:United_States_Statutes_at_Large_Volume_99_Part_1.djvu/151

2 http://www.gao.gov/products/GAO-13-157

3 http://yro.slashdot.org/story/13/01/25/2357219/gao-finds-us-militarys-critical-technologies-list-outdated-useless

4 See Footnote 1 above.

technology velocity, trajectory and disruptive changes to support stakeholders, communities of interest, and other SMEs [subject matter experts]." [5]

In 2014 the sponsor provided initial funding to explore possible implementations of the DTKB concept. This report describes the results obtained pertaining to the use of natural language processing (NLP) technologies to maintain and update a future DTKB.

Components of the DTKB Concept

In coordination with the sponsor the IDA team determined that any DTKB implementation was predicated on the ability to carry out rapid, accurate, and effective subject matter characterization of large collections of existing data. Given the previous experience gained by the IDA team in that area with the IDA Text Analytics (ITA) capability, the IDA team proposed to adopt the ITA as part of the DTKB solution architecture. As the upper part of Figure 1 shows, the proposed DTKB concept envisions the use of the ITA capability to bin the contents of appropriate big data sources into highly homogeneous clusters, i.e., subsets of documents that pertain only to a specific technology, such as phased-array radars, or field-programmable gate array (FPGA) electronics.

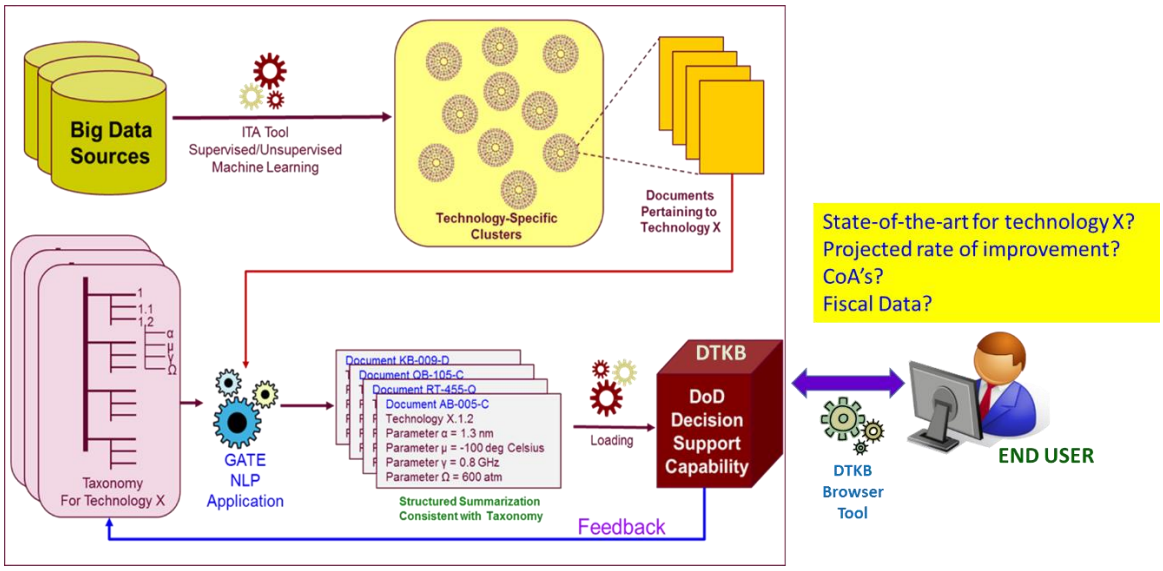


Figure 1. Components of the DTKB Concept

Once the technology-specific clusters have been generated, the next component of the DTKB concept (shown in the lower part of Figure 1) envisions the use of NLP technologies to do the actual extraction of relevant quantitative information. Due to the potentially large number of items that need to be examined for any given technology, as well as the cost

[5] IDA Final Briefing for Project 3669, (informal deliverable)

reduction gained therewith, the proposed DTKB concept focused on alternatives that can support a high degree of automation.

Specifically, the DTKB concept mapped the question of what constitutes the state of the art of a given technology to a taxonomic breakdown of the relevant *technology area*, all the way down to its specific *key parameters*, in a manner analogous to how technology criticality is characterized in the datasheets of the MCTL. These taxonomies are very useful because they can be fed to NLP tools such as the General Architecture for Text Engineering (GATE), an open source suite written in Java, to automatically carry out the needed technology reference identification (TRI), i.e., the extraction of the key parameter values belonging to a given technology from the documents under examination. Once these extracts have been generated they can be placed in adequate repositories, such as Resource Description Framework (RDF) triple stores, to support the appropriate end-user interfaces.

Structure of the Report

Chapter 1 provides an introduction to the DTKB concept and discusses the rationale for the approach taken. Chapter 2 discusses additional NLP techniques that may facilitate the exploitation of large document collections in conjunction with a TRI approach similar to the one discussed in this document. Chapter 3 presents high-level considerations for the solution architecture chosen to carry out a proof-of-principle assessment of the TRI approach at the core of the proposed the DTKB concept. Chapters 4 and 5 present detailed discussions of the ontologies used to power the GATE-based prototype. Chapter 6 discusses the conclusions reached and recommendations made based on the analytical results obtained during the study.

Summary of Conclusions and Recommendations

Based on the analytical results obtained the IDA team reached the following conclusions:

- A fully operational *technical solution* for the proposed DTKB concept can be achieved if adequate funding and sponsor support are provided. The proof-of-principle testing conducted in this study indicates that the specific technologies required to power the proposed DTKB solution architecture have the necessary degree of maturity and are applicable.
- However, none of the available *data sources* tested [6] has a consistent degree of quantitative data content for all the technology areas covered that is sufficient to

[6] The sources tested during the study were: the Defense Technical Information Center (DTIC) (40,000 documents); RDT&E Budget Item Justification Exhibits from 2014 (192 documents); and the Unified Research and Engineering Database (URED) (39,000 database records)

support the highly automated NLP-based extraction procedure envisioned in the proposed DTKB concept.

- In addition, the various *data sources* tested do not use a common vocabulary, thereby making automated cross-comparisons less efficient and increasing the need for human-in-the-loop intervention.

In light of the previous conclusions, the IDA team recommends the following:

- Developing and adopting a *data governance* and *data quality framework* to support the DTKB activity, as well as any other future activity intended to leverage big data concepts and techniques;
- Making a high priority goal the use of *quantitative metrics* in all documents and reports produced for the Federal Government that track technical objectives, accomplishments, trends, etc.;
- Developing and socializing *reference technology taxonomies* across all research activities to ensure the use of common vocabularies to facilitate cross-comparison of results;
- Using consistently and coherently *program element* references (PE numbers) to enable traceability among all data sources, e.g., RDT&E Budget Item Justification Exhibits and URED entries.

Figures

1.

A. The Militarily Critical Technologies List (MCTL)

In 1985, Congress created the requirement for the creation and maintenance of a list of militarily critical technologies.[7] In 2011, funding for the process to update the MCTL began to decrease rapidly, and in 2012, the previous cycle of MCTL reviews and updates by subject matter experts (SME) was effectively terminated.

B. The DoD Technologies Knowledge Base (DTKB)

Since the legal basis for the MCTL remains in force, the Office of the Inspector General (OIG) inquired into the approach planned by the Department of Defense (DoD) to satisfy the congressional mandate. In 2013, the Institute for Defense Analyses (IDA) assessed the various courses of action and briefed the results to the sponsor. Those results were incorporated into the official response to the OIG.

1. Courses of Action

The IDA assessment identified two main alternative courses of action, the second of which had a short-term and a long-term component. The specifics of the proposed courses of action (COA) are described below.

a. Seek Official Relief from the Congressional Mandate

The first COA identified in the IDA report was that DoD ask Congress to eliminate the MCTL requirement because the original purpose of the MCTL has been overcome by events.

The evidence for this is the ability of the export management communities to continue to operate without the MCTL, although perhaps in a less inefficient manner, and the fact that the agencies engaged in export licensing have alternative information sources to satisfy the technical requirements that MCTL was supposed to address.

The benefits of reviving the current MCTL solely for the sake of supporting licensing activities do not appear to warrant the required investment.

---

7 http://en.wikisource.org/wiki/Page:United_States_Statutes_at_Large_Volume_99_Part_1.djvu/151

b. Recreate the MCTL in the Form of a DTKB

The second COA identified in the IDA report was that if DoD were not prepared to seek relief from the Congressional mandate, then it should attempt to satisfy the MCTL requirement by (1) standing up *in the near term* a shared and integrated set of existing information sources constituting a common, dynamic, classified, proprietary, DoD technologies knowledge base (DTKB) that reflects technology velocity, trajectory, and disruptive changes and that is capable of supporting stakeholders, communities of interest, and other SMEs; and by (2) leveraging *in the long term* emerging information technology (IT) techniques (e.g., content understanding software) to develop an IBM Watson-like capability to better answer questions concerning critical technologies and ultimately unburden SMEs from routine issues.

C. Components of the DTKB Concept

In 2014, upon review of the proposed COAs, the sponsor provided initial funding for IDA to carry out a proof of concept assessment of the near-term approach based on a knowledge base to be populated with the contents of existing technical data currently collected by DoD. Because of the time constraints, as well as the funding levels provided, the IDA team decomposed the DTKB concept into four parallel tracks. The details of each track are presented below.

1. Clustering of Big Document Collections by Subject Matter

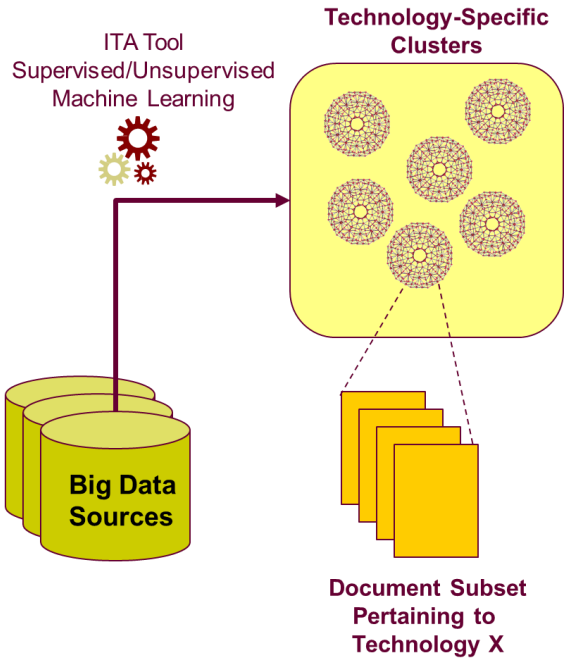


Figure 1-1. Generation of Highly Homogeneous Document Subsets Binned by Technology

Figure 1-1 above shows the first component of the DTKB concept. It leverages the existing IDA Text Analytics (ITA) capability, by extending its functionality to achieve optimal theme discovery, i.e., the ability to separate a collection of source documents into highly homogeneous clusters pertaining to a single technology. This capability would make it possible, with minimal human-in-the-loop (HITL) activity, to process large collections of existing technical reports and bin them by specific technology areas.

2. Automated Generation of Technology Taxonomies

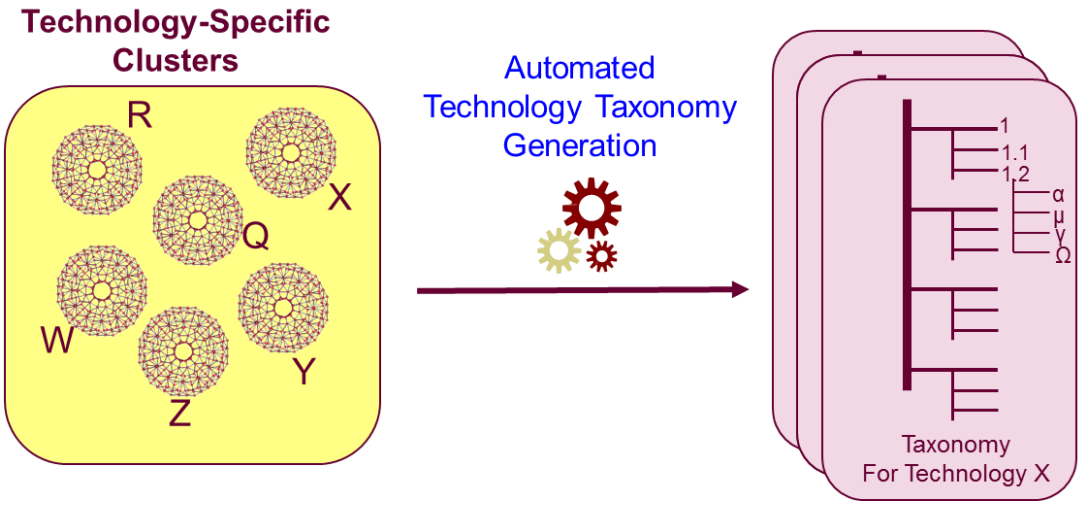


Figure 1-2. Generation of Technology Taxonomies from Technology-Specific Clusters

Figure 1-2 above shows the second component of the DTKB concept, namely, the automation of technology-specific taxonomies that can be used to guide natural language processing (NLP) tools when searching for values of key parameters in a set of documents. Automating this capability would make it possible to keep up with emerging areas of technical interest to DoD without the need for lengthy and costly development of the needed technical vocabularies.

3. Technology Reference Identification (TRI) Using NLP

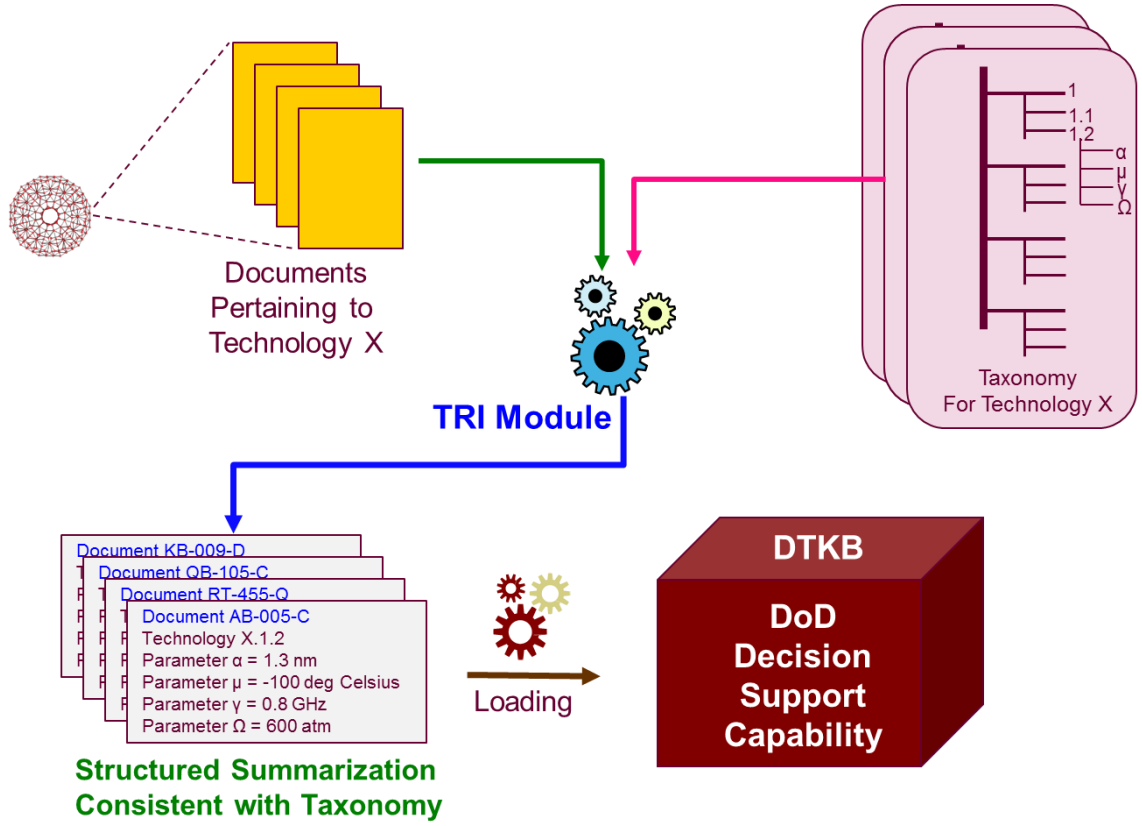


Figure 1-3. Technology Reference Identification Via NLP Technologies

Figure 1-3 above shows the component of the DTKB concept that automates the actual value extraction of key parameters associated with a given technology. By cross-comparing these parameters, it is possible to ascertain the state of the art of a technology, and this in turn enables the end user to make the appropriate decision, for example, regarding export control issues.

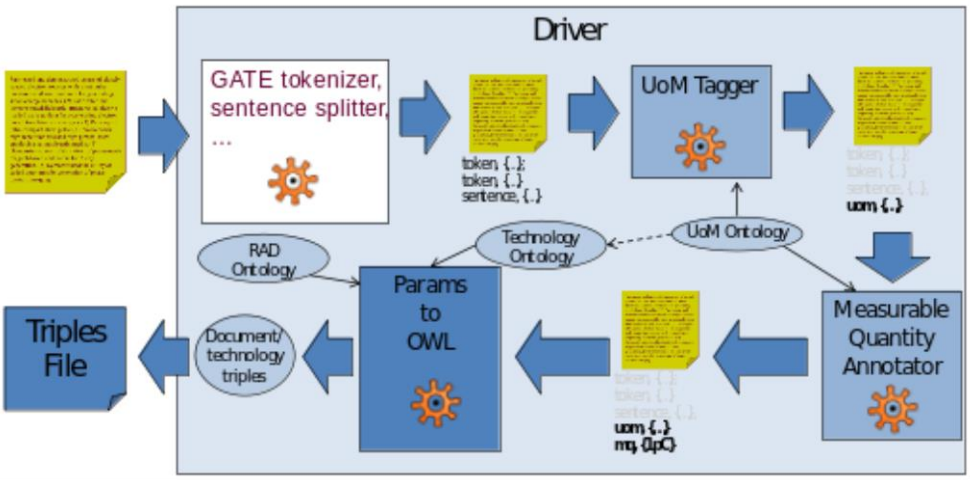


Figure 1-4. Implementation of the TRI Module

Figure 1-4 above shows the internal structure of the TRI module, written in Java, using the architecture of General Architecture for Text Engineering (GATE). Boxes denotes GATE-compatible modules; blue boxes were developed by IDA. IDA-developed modules perform unit of measure tagging ( ), measurable quantity annotations ( ), and the conversion of the results into a Resource Definition Language (RDF) triples file ( ).

4. End User Interface with Natural Language Generation (NLG) Support

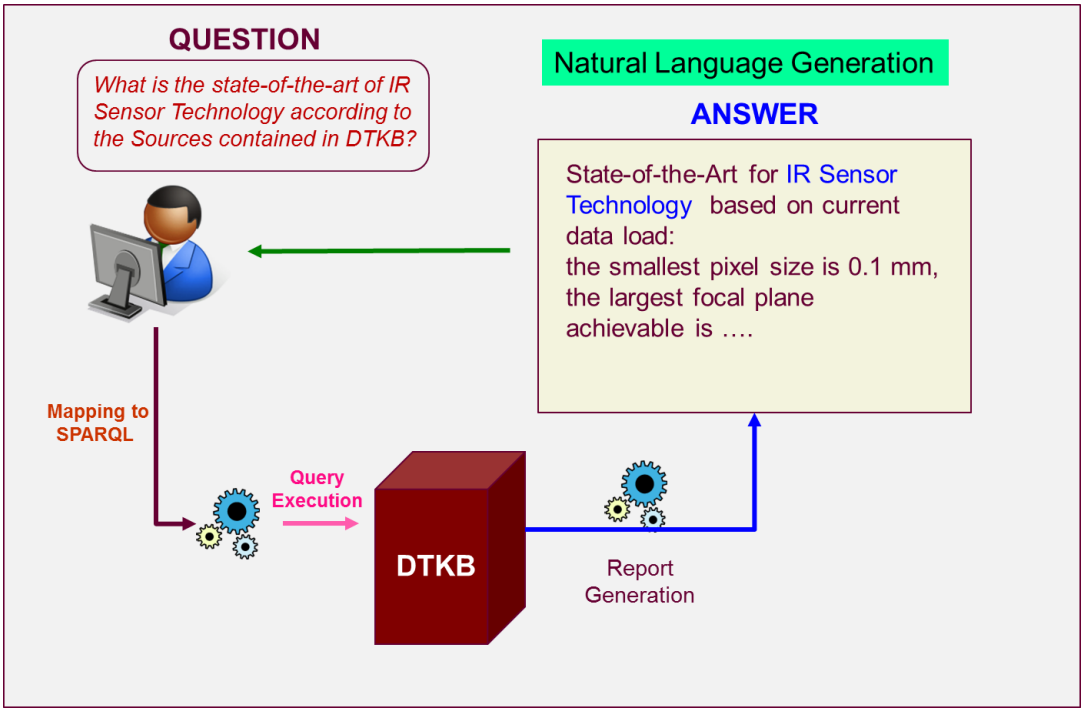


Figure 1-5. End User Interface Using NLG for Decision Making Support

The final component of the DTKB concept is shown in Figure 1-5. Given a particular data content in the DTKB, a user can pose a question regarding the state of the art of a specific technology, e.g., infra-red (IR) sensors. Assuming that the implementation uses an RDF triple store to persist the data, this high-level question can be mapped to a set of SPARQL statements to be executed against the DTKB. The output can then be given to a module that uses NLG to wrap the values found for the key parameters into an easy-to-read narrative that represents the state of the art of the technology.

During this phase of the study, the IDA team generated a notional DTKB data set and loaded it into a Sesame RDF triple store.

Figure 1-6. Login Dialog for the DTKB Querying Tool

Figure 1-6 above shows the dialog box offered to the user to access the application that displays the state of the art of a selected technology.

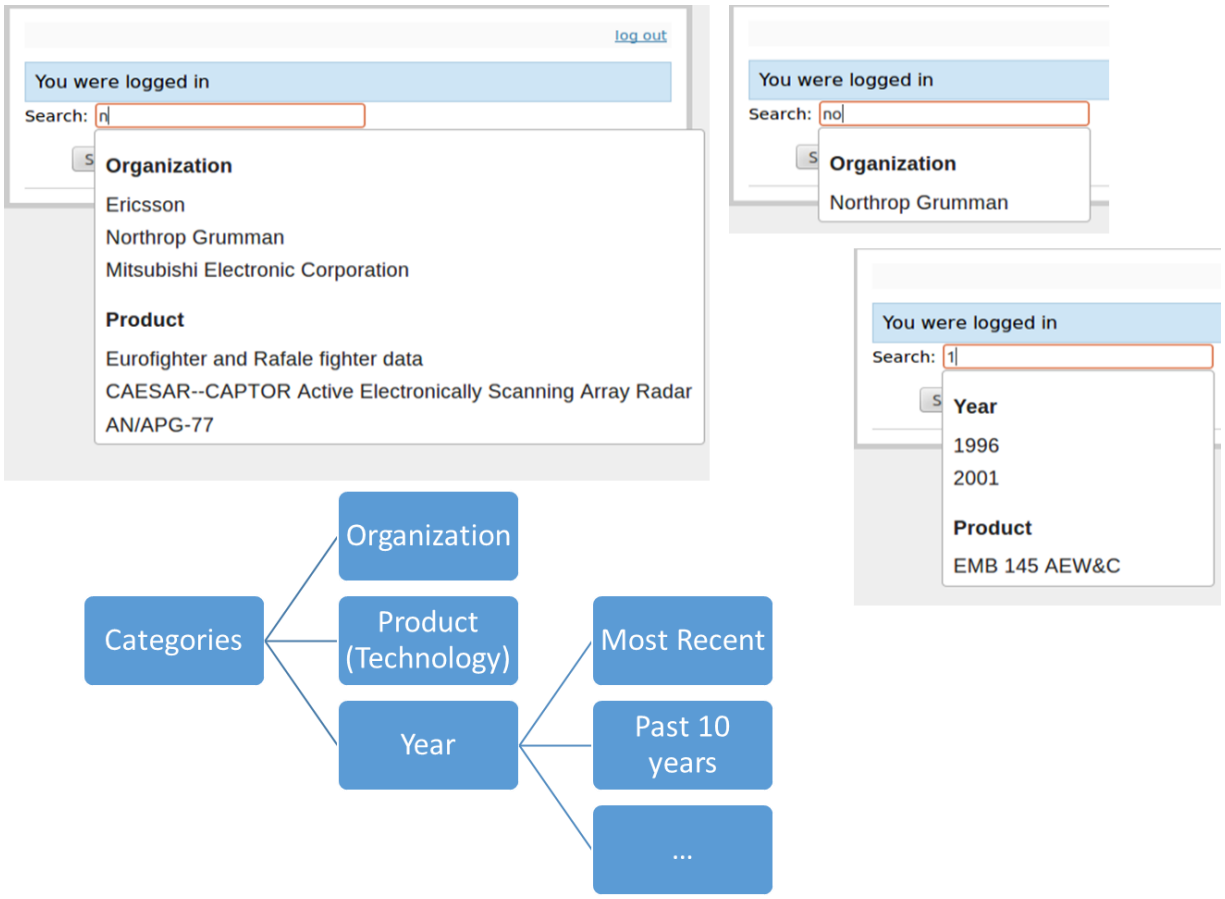


Figure 1-7. Facets supported by the DTKB Browser Tool

Figure 1-7 provides an overview of the various facets supported by the DTKB Browser tool. The implementation provides a rich interactive set of hints to facilitate the retrieval of information most pertinent to the issue being investigated.

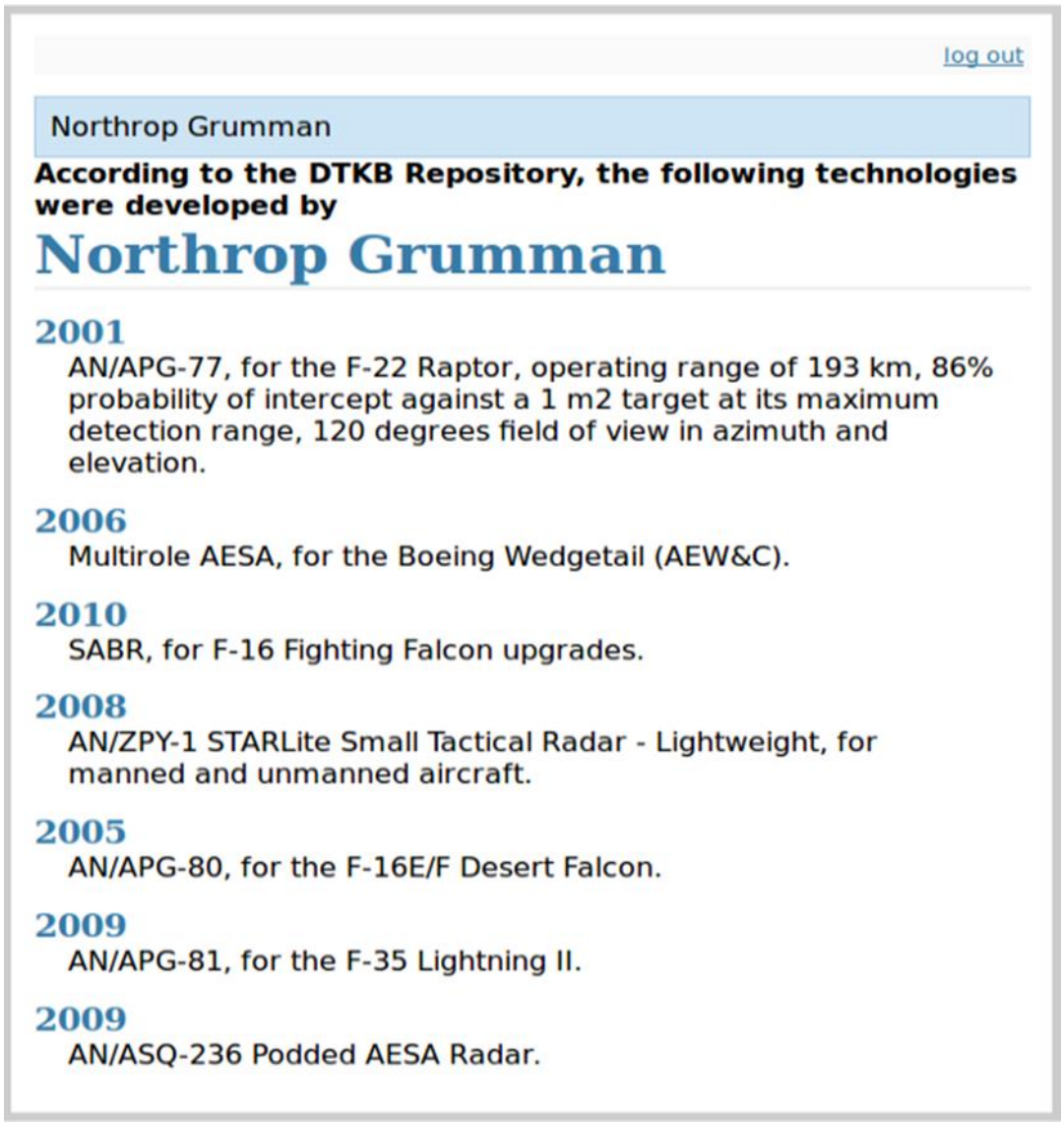


Figure 1-8. Automatically Generated Summarization of Findings

Figure 1-8 shows a notional template that summarizes the contents of the knowledge base when filtered by a specific organization, e.g., Northrop Grumman. The template sorts the entries from oldest to newest and gives a short description of the platform deployed.

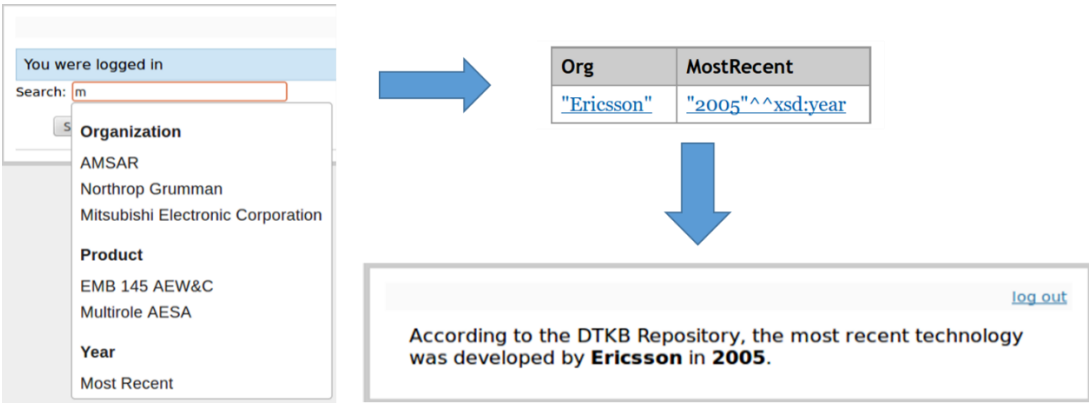


Figure 1-9. Implementation of the NLG Capability

As noted at the beginning of this section, this component of the DTKB concept envisions the use of NLG to facilitate and enrich the answer provided by the DTKB suite of tools. Figure 1-9 above shows schematically how the NLG portion works. The left side of the figure shows the portion of the graphical user interface (GUI) that allows the user to enter the filters. Under the hood these filters are converted into the corresponding SPARQL queries and executed against the RDF triple store. The application takes the results and, using an appropriate template, produces a user-friendly narrative containing the results obtained.

D. Technology State-of-the-Art Browser

Figure 1-5 assumes that the prerequisite filtering of the results has been performed before the NLG module is invoked. The functionality associated with the above-mentioned filtering step is described in this section. To facilitate its design, the IDA team developed a GUI that essentially reflects the workflow associated with deciding which values of the key parameters that characterize a technology constitute the best that have been reported.

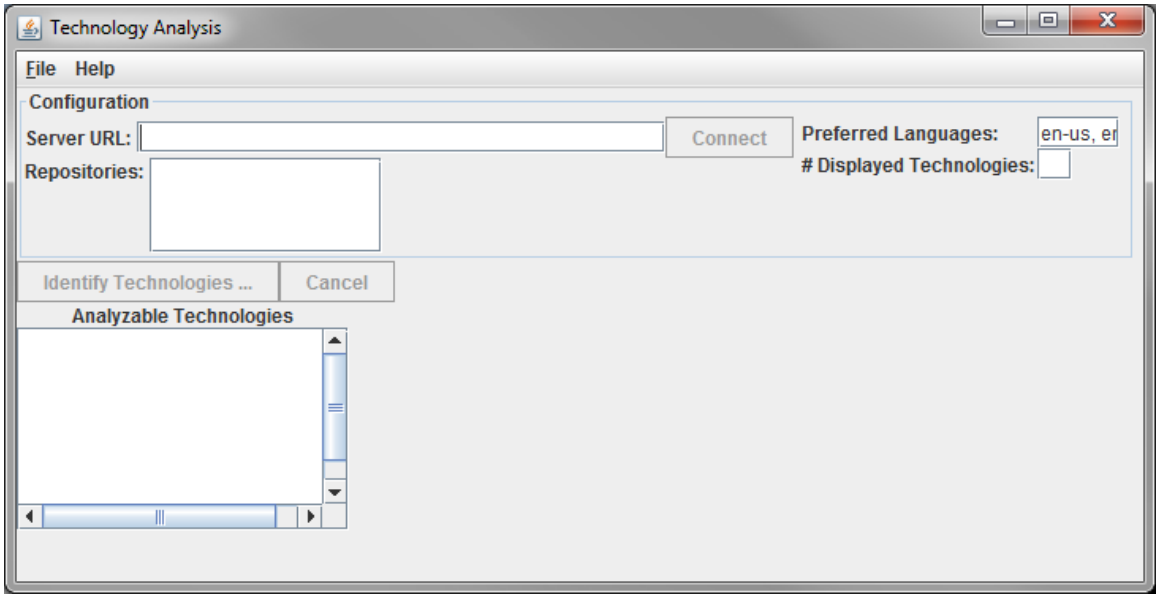


Figure 1-10. Technology State of the Art Browser Start Screen

Figure 1-10 above shows the GUI developed for the Technology State of the Art browser (TSOTAB) application. As shown there, all the functionality is initially grayed out, since it is not connected to a repository.

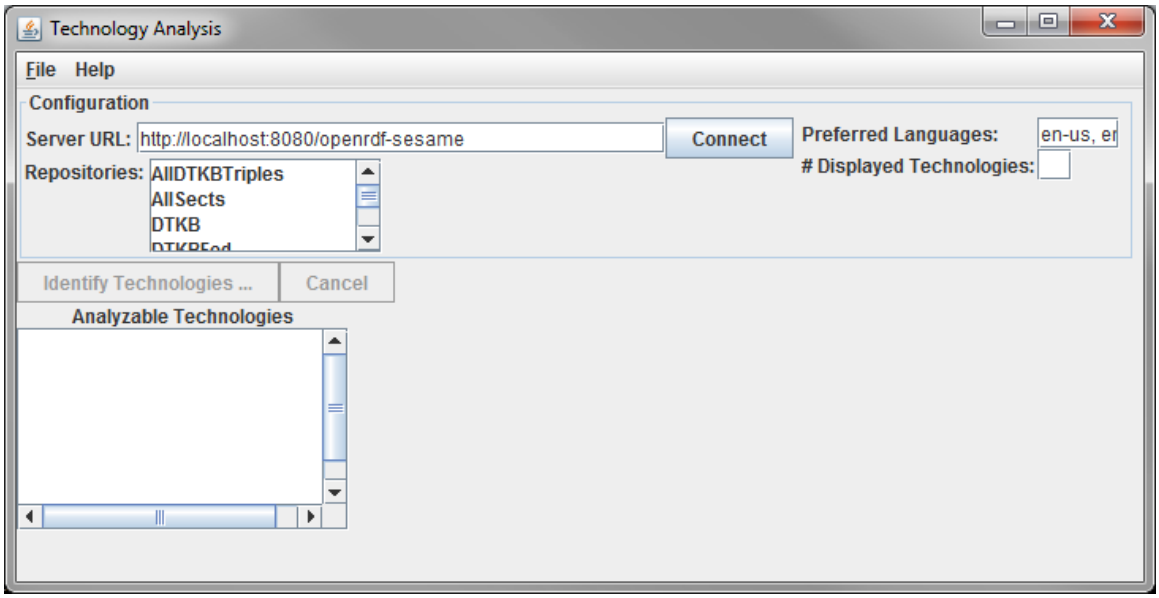


Figure 1-11. Connecting TSOTAB to the DTKB Triple Store

Figure 1-11 shows the state of the TSOTAB application once the user has entered the Uniform Resource Identifier (URL) for the RDF triple store and clicked the button. The Open RDF Sesame application was chosen for testing the DTKB concept. As shown in the figure, Sesame listens on port 8080 when running locally. Once the

connection has been established, the user can select the repositories containing the output of the TRI module. In the example described in this section, all triples are in a single repository named . When the user selects it, the application activates the button. Figure 1-12 shows the state of the TSOTAB application once the appropriate RDF store is selected.

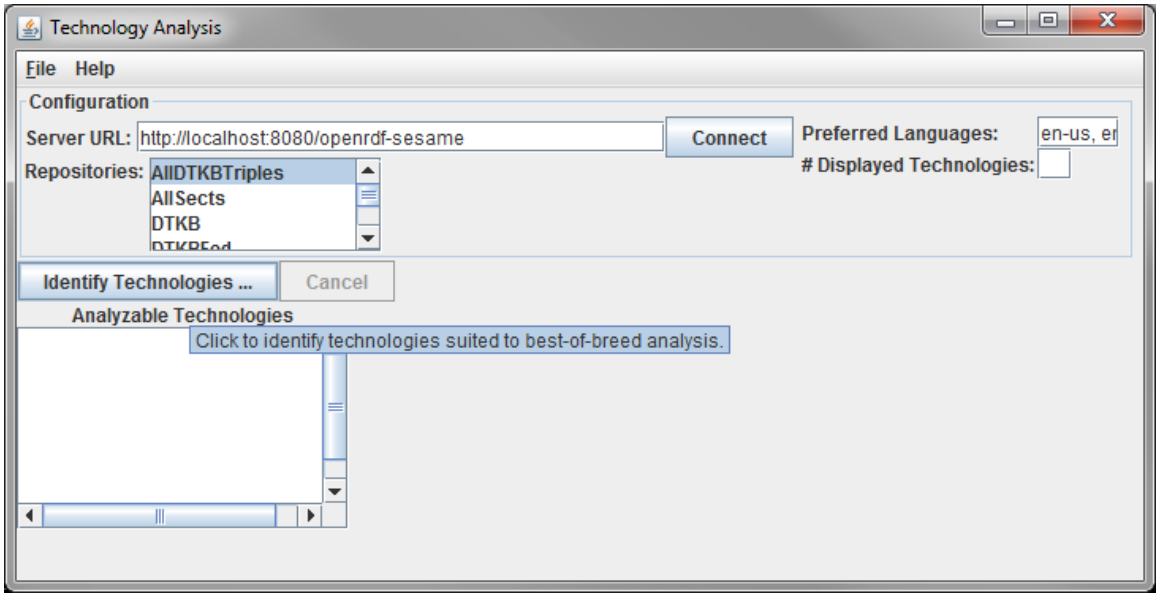


Figure 1-12. Initiation of the Technologies Analysis Step

Figure 1-13 shows the state of the TSOTAB application after it has canvased the contents of the store.

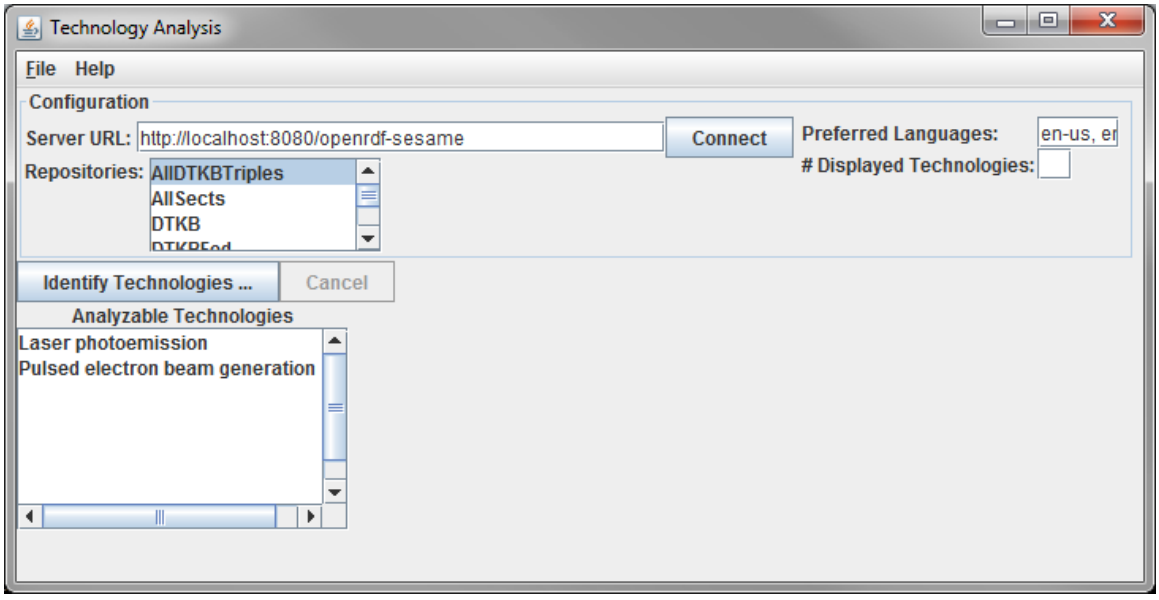


Figure 1-13. Retrieval of Technologies with Quantitative Entries

The interface will show those technologies that have measurable quantitative entries, as defined in the ontologies used by the TRI module. In the example being discussed, the store contains only two analyzable technologies: and . The user next selects one or more of these technologies for analysis.

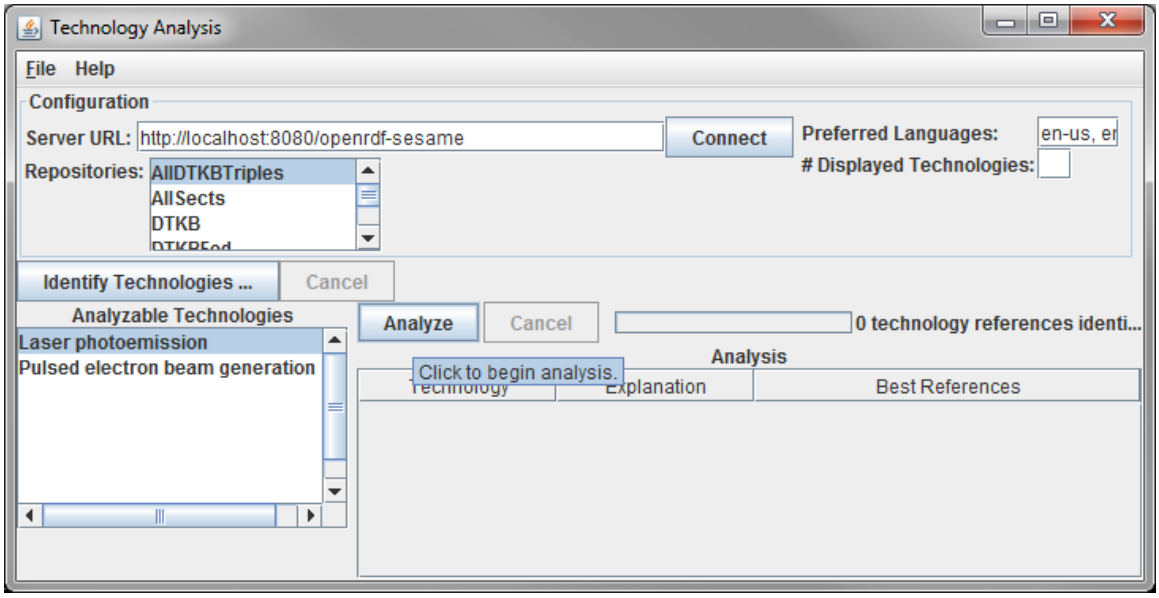


Figure 1-14. Selection and Analysis of Technology State of the Art

Figure 1-14 shows the state of the TSOTAB application once the user has selected an analyzable technology, here Laser photoemission.

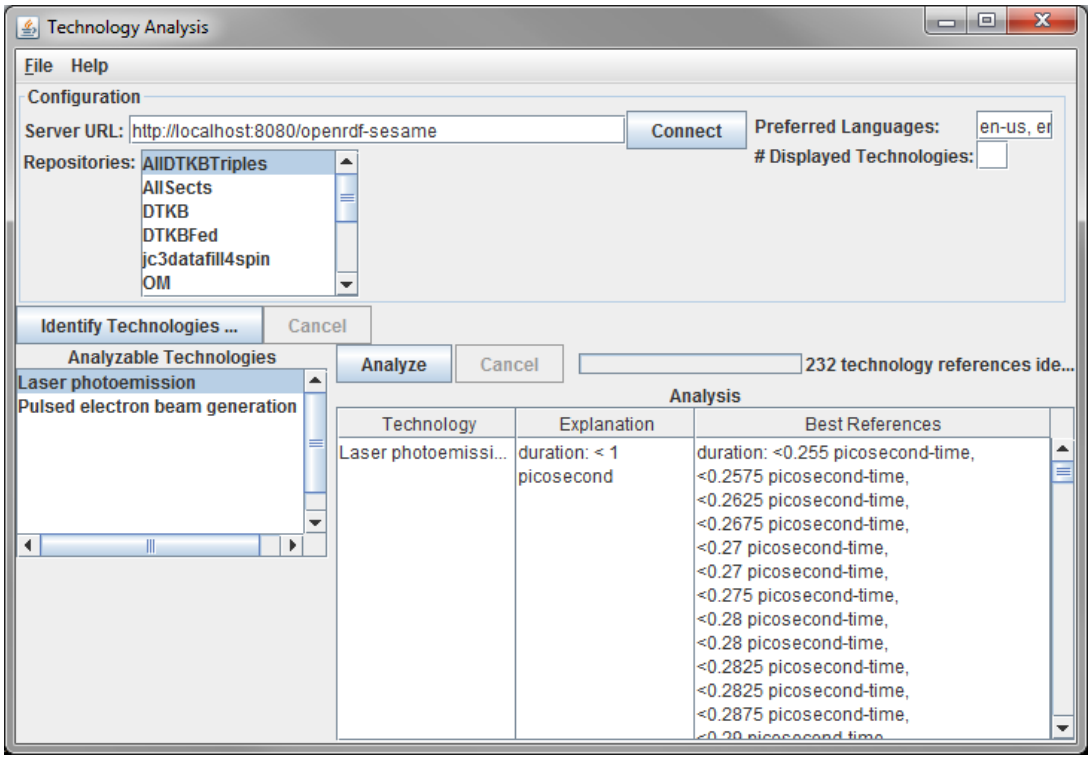


Figure 1-15. Quantitative Values for Technology Key Parameters

The bottom right area of the application displays the widgets used to start analysis and view the results. The user starts the process by pressing the button.

Figure 1-15 above shows the output of the TSOTAB application once the selected technology has been analyzed. The application displays the 232 references to laser photoemission in the DTKB. One parameter of laser photoemission is pulse duration. The application shows the pulse durations of each reference, ordered from shortest to longest. Because the user has not entered a value in the field, the TSOTAB interface will list all values. Note that, according to the ontology implemented for the prototype, the has only one key parameter with measurable values. Obviously, the actual number of parameters with quantifiable measures is larger, and in a production-level implementation, they would be adequately captured in the ontology.

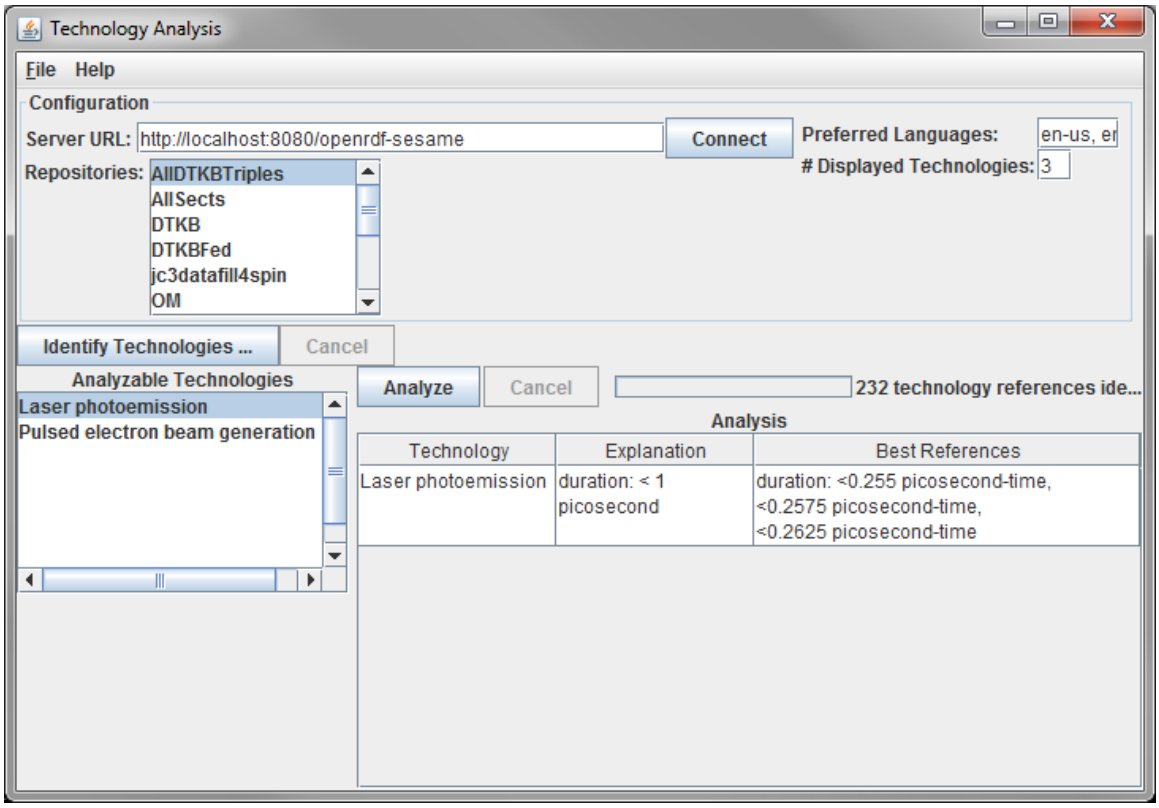


Figure 1-16. Selecting Top Three Entries

Figure 1-16 shows the change in the interface when the field in the TSOTAB application has an entry. Specifically, when the user enters the value 3, the application displays the three smallest pulse durations reported in the source documents. The display shows them appropriately sorted from smallest to largest. In all cases they satisfy the ontology's constraint of being shorter than 1 picosecond.

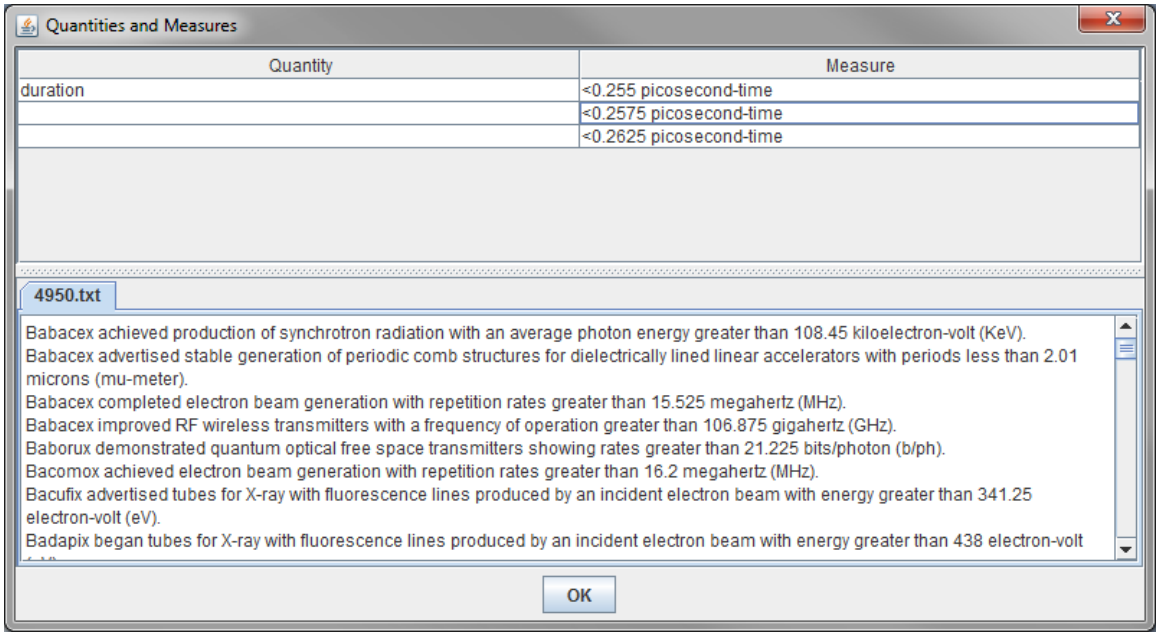


Figure 1-17. TSOTAB Source Text Browsing Capability

The TSOTAB application lets the user search documents to see the references. If the user double-clicks a row in the Analysis table, the TSOTAB application pops up a window showing the analyses in that row. Figure 1-17 shows this window for the results in Figure 1-16. The table at the top shows the measures; it displays the same number of rows as in the field. The window's bottom half displays the documents containing these measures. In the example shown in this section all measures were in a single document, named . That document was a collection of 4,950 sentences containing a computer-generated name for the organization performing the research in 16 different technology areas, one of which was , followed by a verb such as *improved*, *obtained*, *advertised*, etc., and an object clause containing the value of the key parameter associated with said technology.

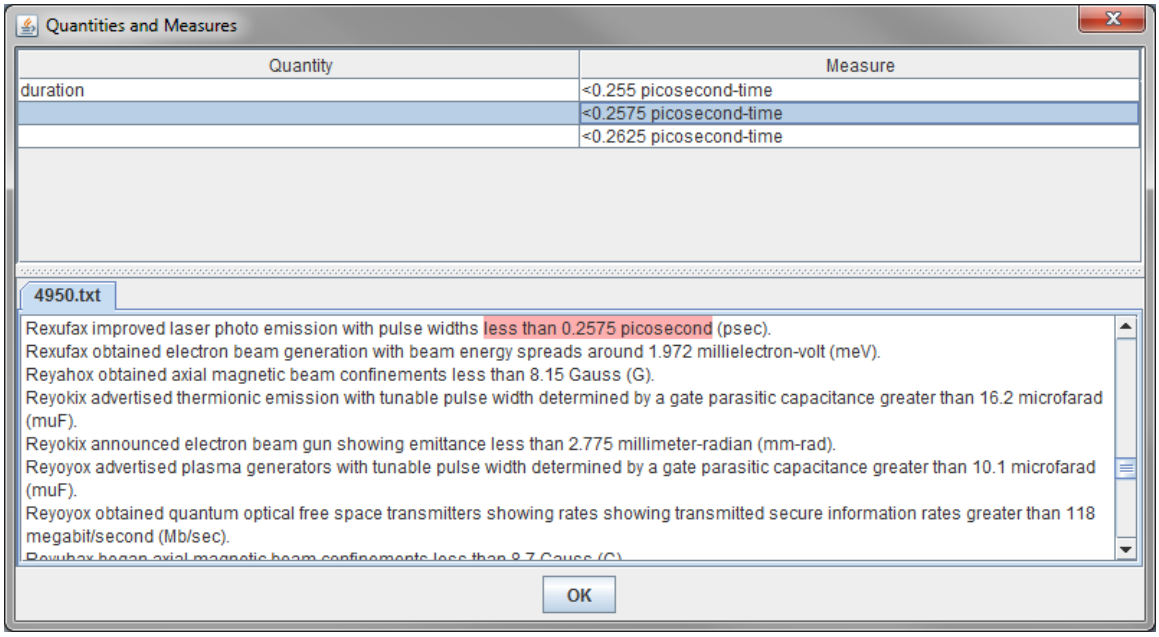


Figure 1-18. Automated Highlighting of Selected Values in Source Documents

Figure 1-18 shows the capability of the TSOTAB application to highlight the value selected within the source documents. When the user selects a row in the top table, the application scrolls to the place in the document where that measure occurs and highlights the reference. This provides contextual information to the user.

2.

A. Introduction

A substantial portion of the information resources necessary to support the DoD decision-making process exists in the form of textual descriptions and narratives. Specific facts, e.g., schedules of planned modernization efforts, responsible agencies for specified activities, detailed technical results pertaining to militarily critical technologies, scope and nature of policies, etc., often require an extensive, expensive, and laborious human-in-the-loop step to convert these data items into actionable information.

Natural language processing techniques can be used to automate the insertion of varying degrees of structure into free text. At a minimum this structuring can provide a useful level of semantic understanding that facilitates analytical activities. For example, parsing a sentence to decompose it into a *subject-verb-object* (SVO) triple can be helpful when performing entity relation (ER) extraction, since entities are normally the actors mentioned in the subject of a sentence. With texts parsed this way, the analysis can concentrate on the subset that comprises just the sentence subjects, thereby reducing the time and effort required to complete the task and, potentially, increasing the accuracy of the results. This type of text structuring also allows the manipulation of textual resources with well-established applications, such as relational databases, and non-Structured Query Language (NoSQL) data storage and retrieval systems, such as RDF triple stores. The latter enables the use of additional semantic resources, such as taxonomies and ontologies, which, in combination with automated reasoning capabilities, can substantially increase the efficiency of the analysts.

A description of an approach for automating the generation of SVO triples intended to be integrated with Component 3 of the overall DTKB concept is shown in the following sections.

B. Description of the SVO Parser Capability

The implemented during this phase of the study is capable of parsing and rearranging input sentences into a regular structure of subject phrase, verb phrase, and object phrase with an additional temporal element, i.e., an adverbial clause that indicates when the actor has accomplished a particular technological breakthrough.

The implemented prototype takes a text file containing the sentences to parse and generates an output text file with the parsed sentences. In addition, the application also generates an output text file into which writes parsed sentences

as RDF triples. The rationale for the last feature is to enable data source traceability across the entire TRI process by maintaining all the information resources in an RDF triple store (implemented either as a single repository or as a set of federated RDF repositories, depending on the specific requirements). The supports various RDF serializations (e.g., Turtle, N-Triples, or RDF/XML notation).

In order to facilitate the parsing of the output sentences, the application can prefix each sentence with indices that give the position of the first character of the verb and object phrases, using non zero-based counting. The current version only provides the index number for the beginning of the object phrase, because the assumption is that this is where the quantitative data pertaining to the key parameters for a given technology are.

The can handle input sentences with a subject, verb, and object elements in either active or passive voice. The application leverages the capabilities of the Stanford Dependency Parser (version 3.5.0), which ideally identifies the correct root verb and at least one correct word in a subject phrase. Once this identification has been done, the can apply a set of heuristics to the dependency tree to generate a *normalized* sentence in active voice.

For example, given the sentence

> Optimal parameters including gas pressure and mixture necessary for < 100 picosecond MPD switching speeds needed for robust survivability in high power electromagnetic fields were successfully determined in 2013 by the researchers at Advanced Research Laboratories.

the will generate as output

> 75:The researchers at Advanced Research Laboratories successfully determined optimal parameters including gas pressure and mixture necessary for < 100 picosecond MPD switching speeds needed for robust survivability in high power electromagnetic fields in 2013.

Sentences with two verbs and two objects are correctly parsed and broken into two separate single verb/object sentences, as shown in the following example:

> The German company Agedum A.G. successfully demonstrated in 2013 classical optical communications over a free space channel with a rate approaching 100 Terabit/s and separately demonstrated a communication system that achieved a photon information efficiency of 12 bits per received photon.

The output produced is:

> 56:The German company Agedum A.G. successfully demonstrated in 2013 classical optical communications over a free space channel with a rate approaching 100 Terabit/s.
>
> 58:The German company Agedum A.G. separately demonstrated a communication system that achieved a photon information efficiency of 12 bits per received photon in 2013.

The heuristics implemented in the can also identify certain trigger words, e.g., *both*, to decompose complex sentences into their individual components, as shown in the following example:

> Kenukix in 2013 improved both its field emission pulse-width control using gate parasitic capacitance of 18.5 micro-farads, as well as its initial testing for reduced field emission pulse-width with proprietary techniques that can operate with gate parasitic capacitance of 25.5 micro-farads.

The produced output is:

> 18:Kenukix improved its field emission pulse-width control using gate parasitic capacitance of 18.5 micro-farads in 2013.
>
> 18:Kenukix improved its initial testing for reduced field emission pulse-width with proprietary techniques that can operate with gate parasitic capacitance of 25.5 micro-farads in 2013.

C. Limitations of the SVO Parser

As noted above, the application leverages the Stanford Dependency Parser library, which occasionally will misidentify the parts of speech contained in a sentence. For example, given the sentence

> Starting in 2013 Taxogux improved field emission pulse-width control using gate parasitic capacitance of 24.9 micro-farads.

the Stanford Dependency Parser identifies '*control*' as the root verb.

Similarly, the Stanford Dependency Parser identifies '*end*' as the subject in the sentence:

> Towards the end of 2013 Nikadox announced reduced field emission pulse-width with proprietary techniques that can operate with gate parasitic capacitance of 15.7 micro-farads.

The Stanford Dependency Parser library also appears to have limitations when confronted with complex sentences such as:

> Fekisox improved in 2012 field emission pulse-width control using gate parasitic capacitance of 11.5 micro-farads and discussed their plans to work on reduced field emission pulse-width with proprietary techniques that can operate with gate parasitic capacitance of 8.5 micro-farads.

In this case the Stanford Dependency Parser library identifies the verb '*discussed*' as the dependent of verb '*using*' instead of '*improved.*' In such cases, a complex sentence is not broken up into two sentences by the .

D. Proposed Next Steps

An in-depth analysis of the heuristics implemented in the Stanford Dependency Parser library is required to improve the performance of the application (currently

around 87% correctly parsed sentences), to reduce the number of misidentified instances to be manually reviewed.

The parser works by initializing itself using a data set created by analyzing a large number of existing natural language documents. The default data set is drawn from a wide range of English documents: technical and nontechnical, fiction and nonfiction. The parser might work better if "trained" on technical documents of the sort likely to contain descriptions of militarily critical, or just plain relevant, technologies.

3.

This chapter describes the nature of a solution to the problem of technology reference identification (TRI) – the third component in the decomposition presented in Chapter 1 above. It focuses on how the available inputs and expected outputs constrain and drive solution architectures. It provides justification for the design decisions IDA made for its prototype identification and analysis system. That the system is a prototype no doubt affected some of the decisions. Someone building a production system in a different environment might reach other conclusions. The reader should keep this in mind when going through the following technical discussion.

A. General Characteristics

Technology references exist in a multitude of documents. We interpret the word "document" as broadly as possible. In theory, anything from a single sentence to a voluminous book, is, for the purpose of this discussion, a document. A document may be textual, pictorial, aural, or a combination of the three. The only requirement is that a document must be uniquely identifiable. The necessity for this constraint is the need to be able to relate a document to other pieces of data, e.g., to be able to make use of a bibliographic reference in a paragraph. A bibliographic reference that cannot be verified and traced back to the document where it was made is of questionable value.

The key aspect of an ideal document is *not* that it be textual, pictorial, or aural, but that it be both structured and exhibit an optimal degree of *atomicity*.[8] In a structured document with high atomicity, the meaning of everything is known. Technology references would be already identified and locating them would require little effort – essentially the implementation of a method to extract them. Ideal documents, unfortunately, constitute a very small portion of the available inputs. (Indeed, producing documents ideally suited for TRI is a secondary objective of the solution architecture discussed in this document.) From

8 In this document we purposely differentiate between *structure* and *atomicity*. A document such as this could be made highly structured by putting its contents into a relational database consisting of tables such as , , , , and . However, this structuring would not necessarily equate with a high degree of *atomicity*, since there would not be a straightforward way to identify which component, e.g., chapter, section, sentence, references a specific subject. In other words, the *meaning* of the content of each component would not be readily available. In contrast, a spreadsheet in which each column corresponds to a well-defined concept, such as , , , , etc., is not only highly structured but also atomic. If one were to find the string *Denver* in the column labeled it would present minimal interpretive difficulties for a user or an application processing such data input.

here on, the team assumed that the documents that will be used as input to the prototype do not have a high degree of atomicity. Furthermore, we confined ourselves to textual (natural language) documents. This limitation is not inherent. Researchers have studied how to extract meaning from pictures, and today's smartphones convert speech to text fairly well. Image identification is a separate topic, however, and speech-to-text systems simply add an extra step to obtaining text. Therefore the team confined its attention to text.

Identifying technology references in text requires parsing the source material. Automated TRI requires scouring online source material, so that sources that do not exist online must be converted to electronic form. Once in electronic form, a document's location and storage format – a single locally accessible file, a cell in a database, a collection of nodes in an RDF triple store, a web page accessible through a URL – scarcely matter; [9] the salient point is that it is available for analysis.

Figure 3-1 depicts this view of the system. On the left are the inputs, divided into three categories: documents accessible through a network, documents stored locally, and documents that, prior to identification and analysis, exist only on paper and must be converted to electronic form. The image of the man by the scanner suggests the extra effort this step entails compared to documents already online. Of course, once a document is scanned, it becomes an online document, as Figure 3-1 shows. That it might be stored non-locally once scanned is irrelevant.

9 It matters insofar as the time it takes to obtain the document affects processing time.

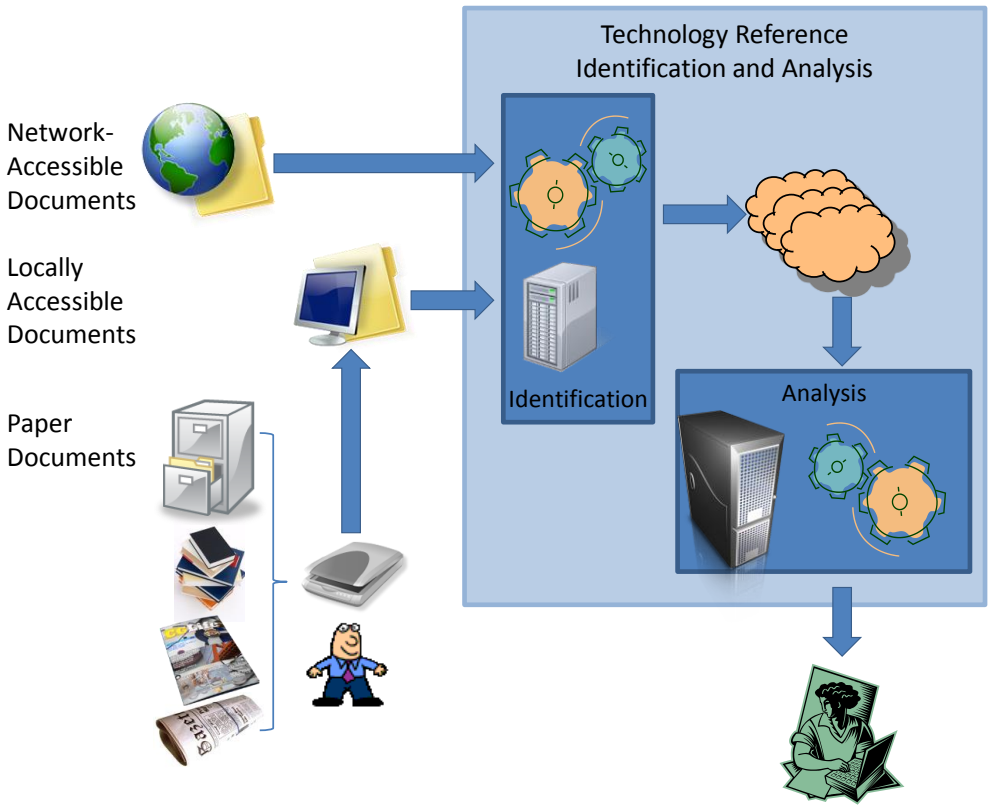


Figure 3-1. System Overview

These electronic documents become inputs to the technology reference identification and analysis (TRI&A) System. The system has two parts: it *identifies* technology references in documents, and it *analyzes* these technology references.

TRI is a knowledge acquisition activity. Its objective is to increase the amount of information known about a document corpus, in particular with regard to technology references. This information is not useful in and of itself. The results of analyses performed on the information, such as how many references total and which year has the most publication year has the most, are the end results people want. In light of the many kinds of analyses conceivable, it makes sense to store the accumulated knowledge so it may be accessed as needs arise. Figure 3-1 shows the documents being processed and stored. In the picture, the storage medium is a cloud, implying the existence of a widely available data store. In practice the knowledge could be stored locally.

The nested box in the lower right depicts an analysis activity. The user in the lower right has invoked an application that accesses the accumulated knowledge and performs some computation that synthesizes the knowledge into a form the user can digest. Figure 3-1 shows only one such application. It is more likely that a system would have many applications, each providing a report of interest to a specific audience.

As the amount of accumulated knowledge increases, it becomes worthwhile to consider storing knowledge about that knowledge. A system based on Figure 3-1 might include an application that feeds its output back into the cloud.

Finally, observe that Figure 3-1 shows two distinct computers. Their use emphasizes the separation between identification and analysis. A production implementation of this kind of system is likely to have a highly distributed implementation.

B. Natural Language Processing

Section A points out that input sources will be documents containing text with a very low degree of atomicity. One can imagine many kinds of text documents that might refer to technology. A non-exhaustive list would include:

- Newspapers,
- Books (anything from textbooks to a Tom Clancy novel),
- Magazines,
- Journal articles, both academic and trade,
- Email,
- Research papers (like this document),
- Websites,
- Standards documents.

An important takeaway from these disparate sources is that one should make few assumptions about writing style or content. A document may have been heavily edited and proofread; equally likely, a single individual may have been its only author and editor.

This paucity of assumptions implies the need for Natural Language Processing (NLP) technology. NLP refers to the capability to transform text into natural-language constructs, e.g., letters, words, punctuation, and sentences. NLP can also parse sentences, generating parse trees and assigning a *part of speech* to every word in the sentence. Parse trees and the assignments that NLP tools generate are not always right, but the tools try, which is often sufficient and certainly better than not trying.

The IDA team, knowing that NLP would be an important component of its identification subsystem, studied several NLP systems. At the time the project began, NLP needs were nebulous. The need to recognize strings in text was clear, e.g., to search for "laser photoemission." That "laser photoemissions" and "photoemission of lasers" are equivalent forms insofar as technology reference identification is concerned strongly suggested that regular expression-based searches would be insufficient. The initial analysis of NLP systems concentrated on their ability to treat these kinds of phrases as equivalent.

The IDA team identified two open-source systems that seemed to provide the requisite capability:

1. GATE, developed and distributed by the University of Sheffield. GATE is an acronym for General Architecture for Text Engineering. Begun in 1995, GATE now comprises a suite of tools for NLP. It can be used as a standalone application, and can also be embedded into an application. GATE claims to be "the biggest open source language processing project, with a development team more than double the size of the largest comparable projects".[10]

2. Apache Unstructured Information Management (UIMA), distributed by the Apache Software Foundation. UIMA is a specification being developed within OASIS [11] for tools that work with non-atomic information. Apache UIMA is an implementation of that specification.

The IDA team concluded that GATE was better suited to its needs. Apache UIMA provides an overarching framework, one that is especially useful on a large project – it provides standards and guidance that developers can tailor. Since the team developing the IDA prototype consisted essentially of a single person, it was not positioned to take advantage of what Apache UIMA offered. GATE, by contrast, had implementations of specific functions that IDA found useful.

C. Accumulated Knowledge: The Case for Semantic Technologies

If the results of technology reference identification are to be available for subsequent analysis, they must be maintained in persistent storage. The technology used to implement this storage strongly influences the overall system architecture. The requirements for persistent storage are as follows:

- It must be able to record technology references. For each reference, it must be able to note the document containing the reference, and the location within the document where the reference occurs.
- It must support queries on known technology references. Information entered must be retrievable.

These two requirements, which effectively state that the storage technology must accept and provide data, may seem self-evident. They have some subtleties, before the elaboration of which, the following additional requirements are noted:

- It must be able to record the "meaning" of a technology. Denoting a technology by a text string is often ambiguous (the canonical example being: "a (military) tank has a (fuel) tank" problem) and therefore insufficient, the more so if the source document corpus is in multiple languages. If a document's use of the

[10] https://gate.ac.uk/overview.html

[11] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uima

word “tank” obviously refers to a fuel tank (e.g., “the tank holds up to 10 gallons”), that fact should be recordable.

- It must be able to integrate disparate bodies of knowledge. This requirement stems from the need to record technology meaning. There are many kinds of technologies, with different vocabularies, models, methods, standards, etc.

1. Requirements Details

The following sections expand on each of these requirements.

a. Recording Technology References

The problem of recording a reference to a technology has two parts. The first is noting the document containing the reference in such a form as to permit its unambiguous retrieval. The second is describing the place within the document where the reference occurs. In their most general forms, both parts present challenges. The first can be addressed using the World Wide Web URL concept. A URL uniquely identifies any document accessible through a server or stored locally (having a “file” protocol). A reference to a locally accessible file becomes invalid if someone tries to access the file from another computer. In Figure 3-1, this would imply that all the input files must be transferred to a server, or that the cloud exists on and is accessed using a single computer, the same one used for technology reference identification and analysis. This was in fact the configuration IDA used in developing its prototype, but it should not be considered realistic in a production environment.

The second part of the problem, describing location, is straightforward in text documents: one simply records the offset of the reference from the document’s beginning. (International character sets complicate the calculations, as the IDA team discovered.) If the problem is generalized to non-textual documents, different approaches are needed. For an audio recording, one might note the duration into the recording that the reference occurs. For an image, a rectangle identifying the relevant part of the image could be used. In this age of multimedia documents, combinations could easily be imagined: a portion of the audio recording that begins at a specified offset, for example. These are interesting possibilities, but IDA only investigated technology reference identification in text documents, so location is not explored further.

Metadata is another area the IDA team also considered but did not investigate. It is easy to imagine noting, along with a technology reference, such items as the time the reference was identified, who identified the reference (useful in a variant of Figure 3-1 with more than one identification system), or its classification level.

b. Querying Technology References

Put simply, information that goes in should be able to get out. The accumulated technology reference knowledge now has the optimal degree of atomicity. Queries should be able to exploit the semantics that such atomicity provides. A good query will be no more complex than the structure it references.

An ideal query is formulated in a domain-specific language, particularly if issued directly by a human. These kinds of languages use terminology and structures familiar to subject matter experts, and are easier to learn than more general-purpose languages.

A query must be able to account for the presence of metadata. If access to certain information is to be restricted based on metadata, a query must be able to express that restriction. Conversely, since metadata presence is often spotty, a query must not fail in its absence.

c. Recording s Meaning

The system's inputs being textual, the idea of recording meaning means there exists a mapping between text strings and the concepts those strings denote. Creating this kind of mapping is at heart of any NLP activity. The ultimate nature of the mapping – its complexity, the kinds of uses that can be made of it – derive from the complexity of the analysis that goes into creating it. The mapping can be syntactic, e.g., whether "tank" is used as a noun or a verb, or semantic, e.g., disambiguating "tank" in some context, or many other things. For technology reference identification, a semantic mapping is necessary. It is important to identify a concept independent of the many strings that might express it. The technology reference identification tool must not care whether one document refers to "x-ray generation technology" and another to "technology for generating x-rays." It must detect that both refer to the same technology.

d. Integrating Disparate Bodies of Knowledge

The necessity of being able to record a technology's meaning independent of its idiographic depiction has far-reaching implications. Technology vocabularies and models tend to be domain-specific and focused on a (relatively) small set of concepts; this makes manageable creating, maintaining, and extending them. Thus the concepts of interest to technology for a military tank (warfare-related) have little to do with those for a fuel tank (hydraulics-related), even though a military tank cannot exist without a fuel tank. It follows that identifying technological advances in fuel tanks for military tanks requires considering not one but two bodies of knowledge.

A technology reference identification system that is not targeted to a specific technology, or set of technologies, needs a mechanism that can include concepts from arbitrary domains and, equally important, can integrate concepts drawn from different

domains. “Integration” means, roughly speaking, the ability to express relationships between concepts from different domains, e.g., that they are equivalent, overlap in some way, or are distinct, or even that nothing is known about their relationship.

2. Semantic Technologies

The IDA team considered these requirements, and elected to implement the persistent storage using semantic technologies; specifically, to use the Web Ontology Language (OWL) and the Resource Description Framework (RDF). These two technologies address the requirements as follows:

- In recording technology references, a document’s location can be expressed as a URL. A reference’s location can be expressed as a pair of RDF triples whose subjects are this URL and whose objects are the reference’s start and end location within the document. URLs are flexible enough to allow access to nontraditional documents – there is no requirement that each document be stored in a single file, just that a server retrieve them as if they are. As noted, URLs with a protocol “file” are not portable, a consideration that must be taken into account for certain implementations. Forbidding certain URLs suffices to guarantee accessibility to any document.
- Given that technology reference identification is a knowledge acquisition activity, integrating reasoning on asserted knowledge is a logical step. IDA was careful to use only forms of RDF that can be expressed in OWL; for example, the two triples described in the previous paragraph can be expressed as data property assertions. Adhering to OWL allowed the prototype to exploit Description Logics reasoning.
- In querying technology references, a collection of RDF triples is a graph, and there are two approaches to accessing that graph. One is to start with a node and traverse its outgoing and incoming edges, following links until a desired goal is reached. The other is to use a graph query language. The former approach is useful for analyzing a specific portion of the accumulated knowledge, the latter for creating summaries based on the entire graph. The former is a programming task, and its viability, or at least effort, depends on the API provided by one’s RDF model implementation. The latter is based on a query language.
- SPARQL is the de facto RDF query language. It is well supported by numerous engines. It was first published as a standard in 2008; a revision appeared in 2013. Although not as powerful as a database query language like SQL, it has clearly been influenced by SQL’s capabilities and recognizes the advantages of many SQL constructs (for example, the SPARQL revision adds aggregate functions and subqueries).

- There are no widely accepted OWL query languages. The Protégé editor supports the Decision Logic (DL) Query language, which lets a user formulate a query using Manchester syntax.[12] DL Query does not appear to be used outside Protégé and is not a recognized standard. Perhaps because OWL can be translated to RDF [13] (meaning that any OWL ontology can be queried using SPARQL), there has not been incentive to standardize DL Query. In any event, SPARQL currently has the most support of all RDF query languages.
- Recording a technology's meaning, the primary purpose of semantic technologies is to enumerate concepts and, to the degree possible, to express their meanings. In RDF and OWL, a concept is identified by its URI. OWL, and to a limited extend RDF, further provides a vocabulary to define the nature of a concept by stating the properties that the concept does and does not possess.
- Regarding integrating disparate bodies of knowledge, one of the strengths of RDF is the ease with which two graphs can be integrated. Integrating relational databases has always proven more challenging than theory might suggest. Two tables from different databases, each named "Person," no doubt represent the same concept, but the keys never match, and one table stores telephone numbers with hyphens whereas the other does not, and other little differences interfere with a neat merge. By contrast, in RDF a person (like everything else) is uniquely identified by a URI, so there is no ambiguity with keys; and if two merged graphs contain multiple telephone numbers for the same individual, a SPARQL query that eliminates duplicates is nothing more than good design.

D. Analysis Technology

Analysis technology supports the development of useful analyses of the accumulated technology reference information. If technology references for a given domain have been accumulated, what support is available to help create reports about those references? Some generic report categories are desirable, such as the number of references or the different documents in which references exist. It is easy to envision useful graphical summary reports, such as a bar chart showing years on the x-axis and number of technology references on the y-axis. This kind of information can be created using SPARQL queries.

Slightly more complicated are reports that require computations. SPARQL supports limited computation – it has arithmetic operators. It has no logarithm function, so, for example, a SPARQL query cannot provide results to depict Moore's law on a log-based y-axis. The IDA prototype includes an analysis tool that orders technologies according to

[12] http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/
[13] http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/

those with the best characteristics, e.g., lowest time, highest energy. SPARQL can sort numbers, but it cannot determine that 10 picoseconds is less than 1 nanosecond – it has no concept of units. The IDA prototype implements this computation. (Of course, one can assert that 1 nanosecond equals 1,000 picoseconds, but the application has to be programmed to make the assertion, which, in terms of effort, is about equivalent to implementing a computation based on ordering.)

Beyond this, one might explore domain-specific approaches to querying. Hydraulic engineers may want answers to specific fuel flow questions, and probably have their own terminology for phrasing those questions. It is certainly possible to conceive of implanting tools that support such terminology and simplify querying for subject matter experts. The IDA team has not explored these avenues.

4.

Technology Reference Identification (TRI) is, self-evidently, dependent on an understanding of the concept of technology. More specifically, TRI relies on being able to examine documents for words, or at least text strings, related to a technology. In IDA's Technology Reference Identification and Analysis system (TRI&A), the concept of a technology and the strings that identify it are encapsulated in an ontology. This chapter describes that ontology, as well as how a subject matter expert uses the ontology to conceptualize a specific technology.

A. Candidate Approaches to Recognizing Technologies

Concepts in the technology ontology express ways in which a document may refer to a technology. Put another way, the technology ontology conceptualizes references to technologies more than it conceptualizes technologies.

The IDA team considered three ways in which a document might refer to a technology:

1. By name. A document might contain the string "laser photoemission technology."
2. By decomposition into lower-level technologies. If a document contains references to several technologies and all of those technologies are known to be necessary components of another technology, it can be reasonably inferred that the document concerns that last technology, even if it makes no direct reference to it. For example, a document that refers to piston technology, lubrication technology, camshaft technology, spark plug technology, and valve technology is probably discussing internal combustion engine technology.
3. By domain vocabulary. Assuming that subject matter experts discuss a technology using a specialized vocabulary, a document that makes use of words from that vocabulary probably refers to the technology.

Techniques implementing each way have strengths and weaknesses. Recognizing a technology by its name is direct and seemingly simple. However:

- It is difficult to anticipate all the ways in which the words that form a technology's name can be arranged, or to account for synonyms. Consider radio frequency wireless transmitter technology. Scanning a document for references to this technology must consider the following points:

- "Radio frequency" is often abbreviated as "RF."
- Domain jargon introduces new word forms: "radio frequency" can be written as "radio-frequency" and "radiofrequency."
- The phrase can be written as "wireless RF transmitter technology" or "wireless transmitter technology in the RF range."
- The phrase "wireless transmitter technology in the ultra-high RF range" is a reference to a specific kind of RF wireless transmitter technology, and it should be recognized as such.
- A document might contain the sentence:

  concern RF wireless transmitter technology.

  which implies that a straightforward text search is inadequate – sentence semantics must be considered.

Decomposition is an interesting possibility for technologies related to complex systems. The IDA team did not explore it, mainly due to the lack of knowledge about technology composition in the available technology descriptions. The approach would likely be based on the existence of a certain percentage of expected technologies, i.e., a document need not refer to every component technology of an internal combustion engine to be about engines. Conversely, a document that refers to piston technology could be about either internal combustion engines or steam engines.

Looking for domain vocabulary concepts is a somewhat indirect approach. It assumes that discussions of technologies will be, for want of a better word, technical. This assumption is consistent with an overarching task goal, namely to assess the state of the art and practice in militarily critical technologies by examining references to them in documents. The IDA team chose to focus on using domain vocabularies.

B. Technology Characteristics

Writings about a technology presumably do more than just name it – they describe it. A technology typically derives from a science or engineering domain, so its descriptions will discuss the technology's observable, measurable, and quantifiable characteristics. Certainly these kinds of characteristics lend themselves to analysis and are most relevant insofar as this task is concerned.

Technologies can be categorized by their characteristics. A technology may be unique in possessing a characteristic, e.g., lithium battery technology is the only battery technology that requires lithium. Alternately, technologies may have the same general characteristics but in quantifiably different ways. Both x-ray generation technology and light bulb technology concern themselves with electromagnetic radiation, but the former is

concerned with wavelengths less than 10 nanometers, the latter between 400 and 750 nanometers.[14] These are two approaches that can be used in technology reference identification.

If a technology is to be recognized by its characteristics, a more precise definition of "characteristic" is needed. All three of the approaches in Section A can be applied, i.e., a characteristic can be described using its name, etc., with analogous advantages and drawbacks. With the emphasis on things that are observable, measurable, or quantifiable, the domain vocabulary of interest is that which concerns physical entities or properties. A characteristic of an RF wireless transmitter is that it emits electromagnetic radiation; this requires a power source and an antenna. Another characteristic is that it generates a certain amount of power. Power is measurable, in watts or some related unit. Some characteristics have quantifiable but not measurable properties. Automotive technology is concerned with four-wheeled (occasionally three-wheeled) vehicles; motorcycle technology is concerned with two-wheeled (occasionally three- or four-wheeled) vehicles.

C. The Technology Ontology

These considerations led the IDA team to develop an OWL-based expression of technology description-related concepts. Figure 4-1 shows the top-level class hierarchy. Those prefixed "tech:" are declared in the technology ontology. Those prefixed "om:" are from the Ontology of units of Measure [15] (hereafter referred to as the OM ontology).

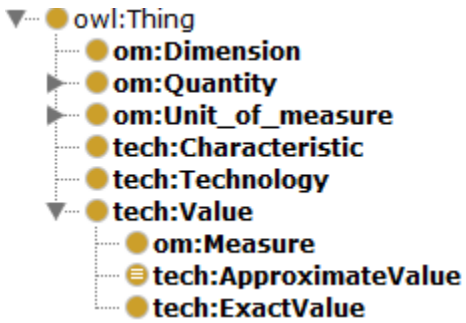


Figure 4-1. Technology Ontology: Top-Level Classes

The technology ontology declares two top-level classes, Technology and Characteristic, whose purpose derives from the discussion in Section B. The ontology declares object property hasCharacteristic, with domain Technology and range Characteristic, to allow the assertion of a technology's characteristic.

---

[14] http://hyperphysics.phy-astr.gsu.edu/hbase/ems3.html

[15] http://www.wurvoc.org/vocabularies/om-1.8/

The ontology can describe a characteristic's properties, using the object property isDescribedBy. This property's domain is Characteristic. Its range is undefined: what a characteristic can describe is unconstrained. In the IDA prototype, characteristics are described by quantities (members of class om:Quantity or one of its subclasses). A quantity is (from the OM ontology):

> and time) of a phenomenon (e.g., a star, food, or a molecule). Quantities are classified according to similarity in (implicit) metrological aspect, e.g., the length of my speedboat and the length of my racing car are classified as length.

A quantity may be expressed as a measure; in OWL terms, OM has an object property om:value, whose domain is om:Quantity and whose range is om:Measure. A measure has a numeric value and a unit (datatype property om:numerical_value and object property om:unit_of_measure_or_measurement_scale, respectively; their domains are both om:Measure, and their ranges are xsd:string and om:Unit_of_measure).

Figure 4-2 depicts these classes and properties. Orange ovals are classes. Blue arrows are object properties, and green arrows are datatype properties.

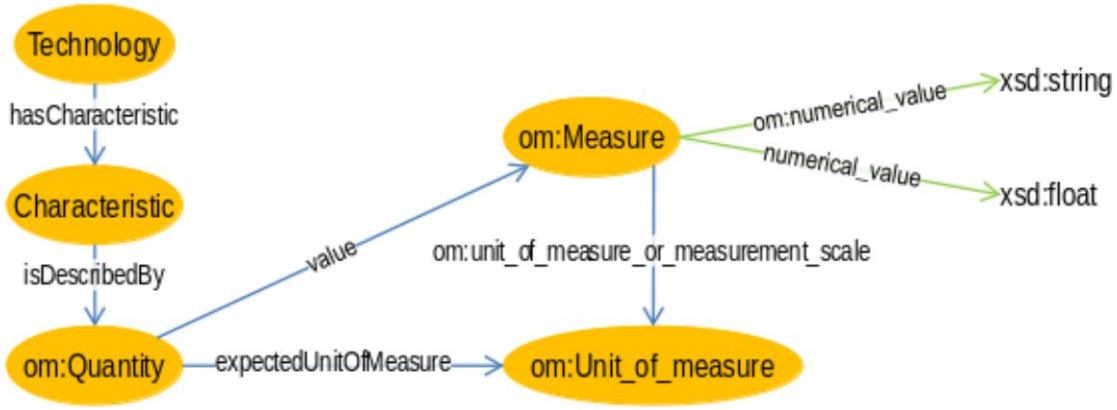


Figure 4-2. Technology Ontology Top-Level Classes and Properties

Figure 4-2 shows two additional properties, numerical_value and expectedUnitOfMeasure. OM's numerical_value property's range is xsd:string. This was a deliberate design decision: OM contains datatype property assertions on numerical_value with objects such as "3 to 5" and "~20.3" (they represent temperature ranges on the Kelvin scale). Lexicographic representation of numbers accommodates such needs. However, some of the analyses the IDA team implemented depend on being able to sort measures, and lexicographic numeric representations don't sort numerically; the string "1" precedes the string "−1", and "10" precedes "2." If reasoning is to account for numeric ordering – and for IDA's technology analysis, it does – string representations of numbers won't work. IDA introduced its own numerical_value datatype property, the range of which is xsd:float. Restrictions on Technology subclasses use this property rather than om:numerical_value.

Property expectedUnitOfMeasure addresses the issue of comparing measures that use different units. One centimeter is less than one meter; unfortunately, there is no way to convey that fact to a Description Logics reasoner. Property expectedUnitOfMeasure supplies a hint that tools can use. If a quantity has a measure whose unit is not the expected unit of measure, a tool can assert that the quantity has an equivalent measure whose unit is the expected unit. OM provides enough information to let the tool convert units.

The technology ontology also contains an annotation property, mustInclude, that is used to provide context to specifications of required characteristics, quantities, and measures. Section D describes how to use it.

D. Extending the Technology Ontology to Represent Specific Technologies

The Technology ontology, as depicted in Figure 4-2, does not model specific technologies but rather defines concepts to do so. It was designed to be extended. It is an ontology; a definition of a technology is a knowledge base. This section covers how one makes use of the technology ontology to populate a knowledge base. It gives examples from the DTKB Technology knowledge base (hereafter DKB) the IDA team created.

Most technology references, although written in singular form, are in fact collective. The phrase "RF wireless transmission technology" refers not to a single technology but to the set of technologies by which radiofrequency wireless transmission can be accomplished. If Technology is the class of all technologies, then "RF wireless transmission technology" is a subclass, not a member, of Technology. In fact, Technology is an ancestor of RF wireless transmission technology. The activity of defining technologies usually uncovers multiple levels of categories into which technologies fit. Figure 4-3 shows the subclass hierarchy into which RF wireless transmitter technology fits.

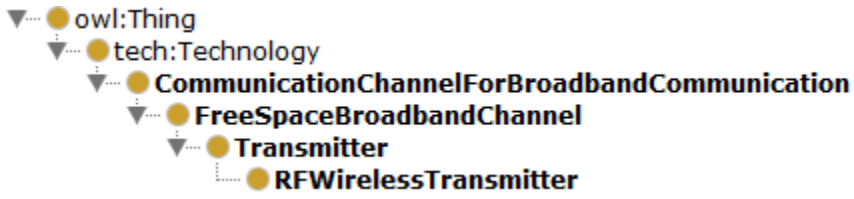


Figure 4-3. A Portion of the Technology Hierarchy

The objective of technology identification is to infer technology class membership from characteristics. A class hierarchy alone doesn't prescribe any characteristics, so the next step is to define them. The IDA team based its characteristics on a technology analysis conducted internally by one of our subject matter experts. It prescribes the following characteristics for RF wireless transmitters:

1. They are fabricated from gallium arsenide, gallium nitride, or indium phosphide.

2. They work by making an electrical circuit impose a signal onto an RF carrier.
3. They emit radio waves, which have a wavelength from 1 millimeter to 100 kilometers. [16]

If each of these characteristics is expressed in OWL, then RF wireless transmitter technology can be defined as the set of individuals possessing them; that is, we can write subclass restrictions on class RFWirelessTransmitter. Accordingly, the IDA team defined subclasses of Characteristic, shown in Figure 4-4:

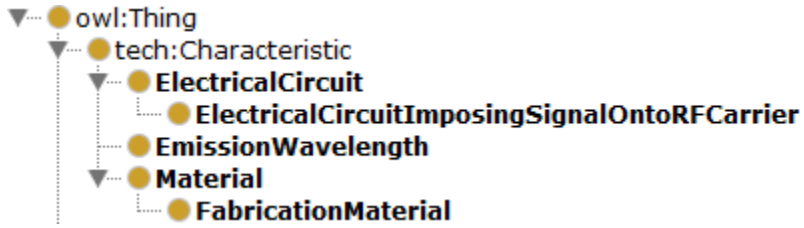


Figure 4-4. Characteristics Related to RF Wireless Transmitter Technology

which allows asserting the following subclass restrictions on RFWirelessTransmitter (using Manchester OWL syntax):

```
hasCharacteristic
    (FabricationMaterial      ( { GalliumArsenide, GalliumNitride, IndiumPhosphide } )
hasCharacteristic       ElectricalCircuitImposingSignalOntoRFCarrier
hasCharacteristic       EmissionWavelength.
```

The three chemical elements in the first restriction are OWL individuals. These individuals have RDFS labels, which are their textual representations in English. The DKB also includes an annotation property chemicalSymbol, used to state the chemical symbol representation. DKB asserts that that the chemical symbol for individual GalliumArsenide is "GaAs," for example.

This characteristic, fabrication material, is observable. Unlike the other two it is not measurable. One can detect whether an electrical circuit has imposed a signal onto an RF carrier, and one can measure the wavelengths emitted by that carrier. The ontology expresses measurability by extending the second and third restrictions. RF wireless transmitter technology not only has an electrical circuit imposing a signal onto an RF carrier, that carrier may be described by the power generated – probably in watts. RF wireless technology not only has an emission wavelength, that wavelength must be between 1 millimeter to 100 kilometers.

[16] http://en.wikipedia.org/wiki/Radio_wave.

These more narrowly focused characteristics describe specific quantities. OM specifies a large set of quantities for measuring the physical world. Figure 4-5 shows the relevant quantities.

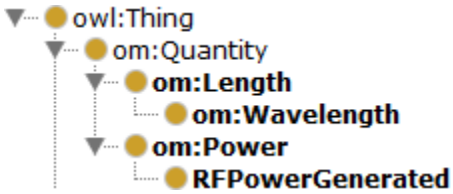


Figure 4-5. Selected Quantities

As Figure 4-5 shows, a quantity has an associated measure. An individual that is a member of class om:Measure denotes a spatiotemporal measure of some phenomenon. An individual is inappropriate for describing characteristics because a characteristic states something that exists throughout time and space. What is needed is a subclass of om:Measure comprising individuals within a certain range. DKB contains such a class for describing wavelength measures, shown in Figure 4-6.

Figure 4-6. Class Defining Length Measures

The following equivalence restriction on this class:

```
(om:unit_of_measure_or_measurement_scale          om:metre)
    (numerical_value some xsd:float[>= 0.001f, <=1.0e5f])
```

Means that a DL reasoner will infer that any individual whose unit of measurement is meters, and whose value is asserted to be between 10-3 and 105, is a member of the class LengthBetweenOneMillimeterAndOneHundredKilometers. The restriction must be an equivalence restriction; a subclass restriction is too weak to allow the inference. The units are normalized to meters. The restriction can be stated using millimeters and kilometers, but at the cost of introducing a complicated Boolean expression.

With these new classes in place, the second restriction (an electrical circuit imposing a signal) becomes:

```
hasCharacteristic
    (ElectricalCircuitImposingSignalOntoRFCarrier
            (isDescribedBy
            (RFPowerGenerated       (expectedUnitOfMeasure          om:watt))))
```

which can be represented visually (omitting some intermediate classes) as in Figure 4-7.

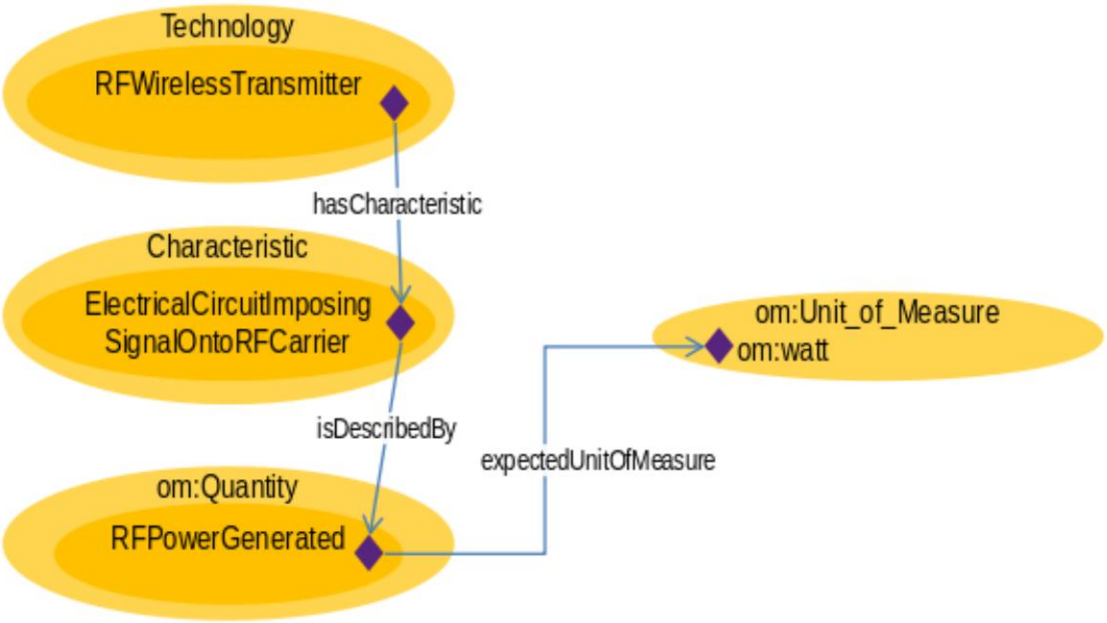


Figure 4-7. RF WirelessTransmitter Must Generate RF Power

The third restriction becomes:

```
hasCharacteristic
  (EmissionWavelength
        (isDescribedBy
     (om:Wavelength
           (expectedUnitOfMeasure        om:meter)
           (om:value        LengthBetweenOneMillimeterAndOneHundredKilometers))))
```

the picture for which is shown in Figure 4-8.

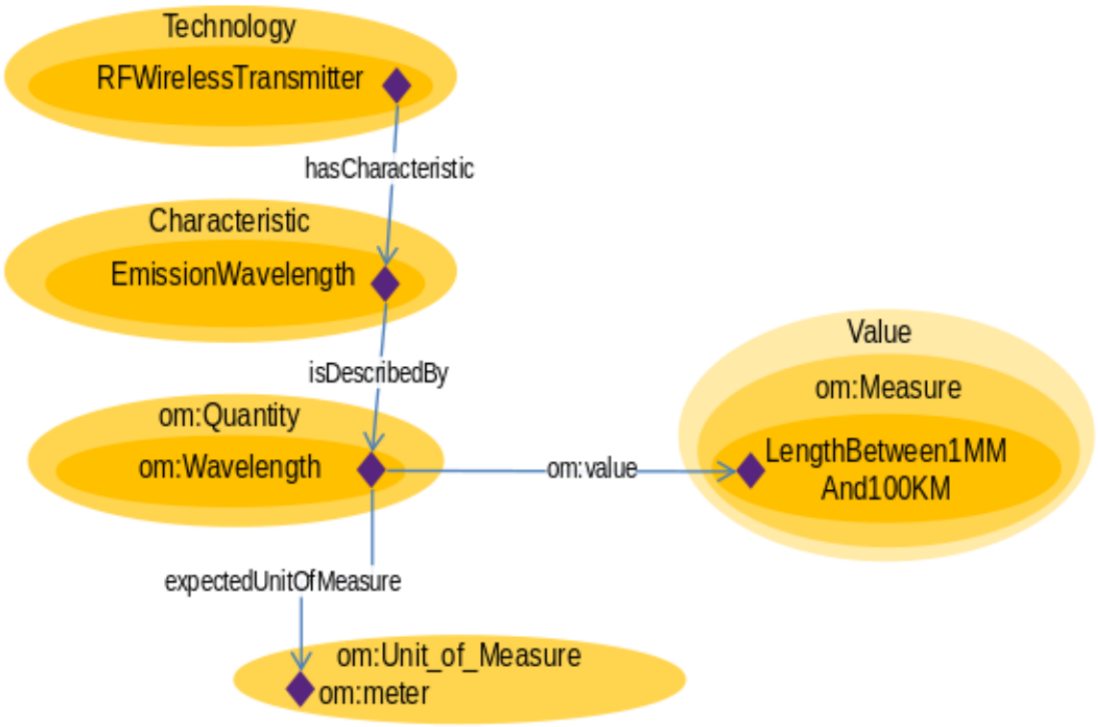


Figure 4-8. Emission Wavelength Must be in RF Range

Figure 4-7 and Figure 4-8 differ in that only the latter has a measure for its quantity. The IDA team did not find any numerical restrictions on power generation. This has implications for technology reference identification, as Section E discusses.

A wavelength can be anywhere between 1 mm and 100 km – a broad range that covers many likely measures of length. To prevent common measures from being misinterpreted as technology references, a technology knowledge base may require that a measure be considered in context. The knowledge base may specify that the sentence containing the measure must include certain keywords for the measure to be a reference to a technology. This is done using the mustInclude annotation property. It is asserted as an annotation on a technology subclass restriction. In OWL terms, the annotation is an axiom whose source is the technology subclass; property is mustInclude; target is the subclass restriction; and which asserts one or more instances of mustInclude, the values for which describe the keywords. Suppose only sentences containing the word "wavelength" are to be considered as candidate references to RF wireless transmitters. The assertion (derived from the third restriction)

```
RFWirelessTransmitter rdfs:subClassOf
  hasCharacteristic
    (EmissionWavelength
          (isDescribedBy
       (om:Wavelength
             (expectedUnitOfMeasure        om:meter)
             (om:value
            LengthBetweenOneMillimeterAndOneHundredKilometers))))
```

is annotated such that:

- RFWirelessTransmitter is the annotated source.
- rdfs:subClassOf is the annotated property.
- The restriction is the annotated target.
- The annotation asserts "mustInclude "wavelength"^^xsd:string".

The value of the annotation assertion may be any RDF literal, or a Uniform Resource Indicator (URI). If it is a URI, that URI's label (if one exists) is used as the basis for required words.

The rules for specifying technologies in terms of the Technology ontology may be summarized as follows:

- Technologies are specified as descendants of class Technology. Insofar as possible, technologies should be organized in a hierarchy.
- A technology's semantics are specified using class restrictions on the characteristics the technology possesses.
- These restrictions should be as specific as possible:
  - They should include individual characteristics known to be inherent to the technology.
  - They should reference quantities that describe the characteristics.
  - They should include the expected unit of measure for the quantity, if one is known.
  - They should include limit valid measures for the quantity, if limitations are known.

Developing a technology knowledge base requires both subject matter expertise (regarding the technologies in question) and experience using ontologies. Conceivably the latter can be packaged and automated, meaning the effort need only involve subject matter experts. The IDA team did not investigate this topic.

E. Using a Technology Knowledge Base

Section D explains how to extend the Technology ontology to define technology knowledge bases. This section explains how the IDA team expects a technology knowledge base to be used in technology reference identification.

Inputs to technology reference identification include a technology knowledge base and a corpus. Technology reference identification involves scanning documents in the corpus, looking for:

- Words or phrases that appear to be technologies, characteristics, or quantities in the technology knowledge base,
- Text sequences that appear to be measures using units in the technology knowledge base.

Consider the following paragraph from a hypothetical document:

> In 1999, Widgets Incorporated released its first-generation Gallium Arsenide-based RF wireless transmitter. It emitted signals with a wavelength of 8.2 millimeters.

The first sentence contains a phrase, "RF wireless transmitter," that is the name of a technology (more precisely, it is the value of the label of class RFWirelessTransmitter) and a chemical name that is one of the recognized values for the fabrication material characteristic. The second sentence contains the word "wavelength," which is the name (and label) of a subclass of om:Quantity, and the string "8.2 millimeters," which is a measure.

All four items suggest that this document refers to RF wireless transmitter technology. As Section A mentions, none of the strings taken by itself conclusively proves the reference, but each is suggestive enough to suggest that someone interested in RF wireless transmitter technology should examine the document.

Recognizing that the 8.2-millimeter measure is a characteristic of RF wireless transmitter technology is a computational problem. The DKB specifies the possible range of measures. Someone has to compute that 8.2 millimeters is within that range. A DL reasoner can make the inference, but only if the knowledge base is amended with an equivalent measure whose units are meters rather than millimeters. The reasoner then will infer that the measure is of type LengthBetweenOneMillimeterAndOneHundredKilometers. Given that class's relationship to RFWirelessTransmitter (specified by the restriction), a technology reference identification tool can infer that the measure is used when discussing RF wireless transmitters. As explained at the end of Section D, this inference is subject to the sentence also containing "wavelength" (which it does), ensuring that the sentence "Alan is 1.8 meters tall" is not interpreted as a reference to an RF wireless transmitter.

5.

Technology reference identification yields a collection of descriptions of documents that contain references to technologies. This chapter discusses the representation of those descriptions.

A. Ontology Elements

Descriptions are represented using concepts expressed in the Range-Annotated Document (RAD) ontology. The RAD ontology is a small ontology whose purpose is to standardize the concepts used to represent descriptions. The Technology Reference Identification tool generates output consistent with the ontology. The Best Technologies tool assumes that it is accessing a knowledge base containing triples consistent with the ontology.

The RAD ontology introduces two classes, and makes use of two OM classes. The two classes it introduces are:

1. Document. This class denotes the set of document individuals. A document individuals is something that can contain annotations.
2. Annotation. An annotation derives from the natural language processing concept of a stretch of a document's content. An annotation is associated with exactly one document. It has a start and an end, integer values that are offsets denoting the annotation's start and end.

The RAD ontology uses OM classes Measure and Quantity. An annotation denotes a quantity; that is, the existence of an annotation denotes that the document contains a quantity. A quantity has a measure as its value. Object properties express these relationships.

Figure 5-1 depicts the RAD ontology. It shows the classes and properties described in the previous paragraph. It also shows property denotesMeasure, which relates Annotation and om:Measure. The property is a chained property.[17] It is inferred by the existence of:

```
:annotation rad:denotesQuantity :quantity .
:quantity om:value :measure .
```

[17] http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Object_Subproperties

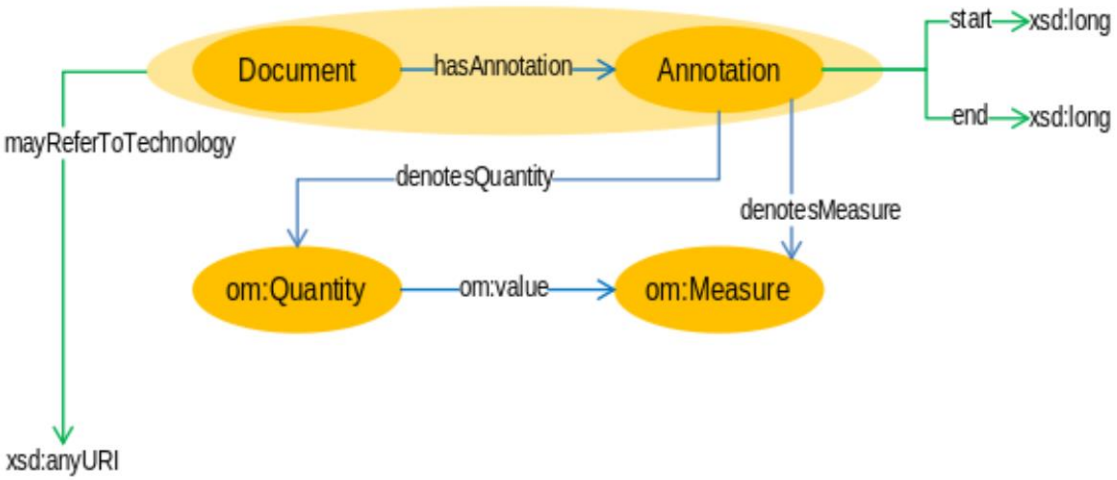


Figure 5-1. The Range-Annotated Document Ontology

or formally, in Manchester syntax:

rad:denotesQuantity om:value rad:denotesMeasure.

Inferring denotesMeasure is convenient for SPARQL queries, which can eliminate the intermediate reference to an om:Quantity individual.

Technology references are expressed using datatype property mayReferToTechnology. The property's domain is the union of Document and Annotation. The ability to associate a technology reference with an entire document provides flexibility for cases where the exact location of a technology reference is vague. (It is not currently used.)

Property mayReferToTechnology's range is xsd:anyURI. The value of an assertion involving this property is expected to be the URI of a technology. Recall that a technology is specified as a subclass of Technology (see Section 4, Section D). A document is an OWL individual. In an OWL property assertion p(individual, value), value can never be an OWL class. If p is an object property, it must be an individual. If p is a datatype property, it must be an RDF literal. Using a datatype property whose range is xsd:anyURI is the best way to refer uniquely to a technology.

Figure 5-1 does not show it, but denotesMeasure and hasAnnotation have inverses. These inverses are anticipated to be useful in SPARQL queries.

These concepts form the minimal information needed to express technology references. Tools are permitted to add properties for measures and quantities.

B. Example Queries

A knowledge base populated with technology references consistent with the RAD ontology can be queried to obtain reports on the references. The following are examples of SPARQL queries that may be useful.

The following query identifies all documents that may refer to a technology:

```
@PREFIX : < http://www.ida.org/nlp/range-annotated-document#>
SELECT DISTINCT ?document
WHERE {
  ?document :hasAnnotation _:annotation .
  _:annotation :mayReferToTechnology ?technology
}
```

The following query identifies documents and the number of technology references each contains:

```
@PREFIX : < http://www.ida.org/nlp/range-annotated-document#>
SELECT ?document (COUNT(?technology) AS ?nTechRefs)
WHERE {
  ?document  :hasAnnotation _:annotation .
  _:annotation :mayReferToTechnology ?technology
}
GROUP BY ?document
```

The following query provides a complete report on technology references identified, assuming a reasoner has been used:

```
@PREFIX :         < http://www.ida.org/nlp/range-annotated-document#>
@PREFIX om:     <http://www.wurvoc.org/vocabularies/om-1.8/ >
@PREFIX tech:   <http://www.ida.org/nlp/technology#numerical_value>
SELECT ?document ?technology ?start ?end ?unit ?value
WHERE {
  ?document  :hasAnnotation _:annotation .
  _:annotation :mayReferToTechnology ?technology ;
                :denotesMeasure _:measure ;
                :start ?start ;
                :end ?end .
  OPTIONAL {
      _:measure   om:unit_of_measure_or_measurement_scale ?unit ;
                  tech:numerical_value ?value .
  }
}
```

This query slavishly adheres to the concepts expressed in the RAD ontology and makes information regarding the measure optional.

6.

A. Introduction

As briefly described in Chapter 1, the proposed DTKB concept comprises four components. Both Component 1, *clustering of big document collections by subject matter*, and Component 2, *automated generation of technology taxonomies*, leverage the extensive experience gained by the IDA team in the development of the ITA capability. The results obtained during the study indicate that these two components of the DTKB concept do not represent a technical risk.

Similarly, Component 4, *end user interface with NLG support*, leverages well-established frameworks for data persistence and web solution development, as well as the extensive experience the IDA team has using semantic technologies. Specifically, the results obtained using Open RDF Sesame as the RDF triple store for the extracted summarizations, and of Flask, a Python micro-framework for web applications, to create a highly interactive web-based interface, indicate that there are no technical risks in using these technologies in a DTKB implementation.

From a technical point of view, Component 3, *technology reference identification using NLP*, was by far the most challenging of the DTKB components. The IDA team adopted GATE as the NLP framework to power the automated extraction of quantitative key parameters that characterize the state of the art of a technology due to its robustness and large user base, as well as its support for the use of taxonomies and ontologies to perform entity relation extractions. Unfortunately, not all GATE modules are well documented and a fair amount of trial and error was required to get them to work as desired. The results obtained during the study indicate that the approach chosen does not represent a high risk. Nevertheless, it would be prudent to explore other approaches to minimize the risk involved in a future implementation.

B. Preliminary Conclusions

Based on the analytical results obtained, the IDA team reached the following conclusions:

- A fully operational *technical solution* for the proposed DTKB concept can be achieved if adequate funding and sponsor support is provided. The proof-of-principle testing conducted in the study indicates that the specific technologies required to power the proposed DTKB solution architecture have the necessary degree of maturity and are applicable.

- However, none of the available *data sources* tested, i.e., DTIC (40,000 documents), R2 Exhibits from 2014 (192 documents), URED (39,000 database records) has a consistent degree of quantitative data content for all the technology areas covered sufficient to support the highly automated NLP-based extraction procedure envisioned in the proposed DTKB concept.
- In addition, the various *data sources* tested do not use a common vocabulary, thereby making automated cross-comparisons less efficient and increasing the need for human in the loop intervention.

C. Preliminary Recommendations

In light of the previous conclusions the IDA team recommends the following:

- Proceeding to the next phase of the DTKB assessment in order to obtain a definitive answer regarding the best way to satisfy the MCTL requirement.
- Developing and adopting a *data governance* and *data quality framework* to support the DTKB activity, as well as any other future activity intended to leverage big data and content understanding technologies.
- Adopting a realistic schedule for a DTKB implementation and providing adequate funding and support.
- Mandating across the DoD the use of *quantitative metrics* when reporting all technical objectives, accomplishments, trends, etc.
- Developing and socializing *reference technology taxonomies* across all DoD research activities to ensure the use of common vocabularies to facilitate cross-comparison of results.
- Mandating across the DoD a consistent and coherent use of *program element* references (PE numbers) to enable traceability among all data sources (e.g., RDT&E Budget Item Justification Exhibits (a.k.a. R2 Exhibits) and URED entries).

The rationale for the above recommendations is the patently obvious fact that machines cannot extract content where there is none. Substantive productivity gains and cost reductions are predicated on the existence of a disciplined approach for reporting and sharing technical data that ensures maximum quantitative content and common vocabularies.

API	application program interface
COA	course of actions
DKB	DTKB Technology knowledge base
DL	Decision Logic
DoD	Department of Defense
DTIC	Defense Technical Information Center
DTKB	DoD Technologies Knowledge Base
ER	entity relation
FPGA	field-programmable gate array
GAO	Government Accountability Office
GATE	General Architecture for Text Engineering
GUI	graphical user interface
HITL	human-in-the-loop
IDA	Institute for Defense Analyses
IR	infra-red
IT	information technology
ITA	IDA Text Analytics
MCTL	Militarily Critical Technologies List
NLG	natural language generation
NLP	natural language processing
NoSQL	non-Structured Query Language
OIG	Office of the Inspector General
OM	Ontology of units of Measure
OWL	Web Ontology Language
PE	program element

R2	RDT&E Budget Item Justification Exhibits
RAD	Range-Annotated Document
RDF	Resource Definition Language
RF	radio frequency
SME	subject matter expert
SPARQL	RDF Query Language
SQL	Structured Query Language
SVO	Subject Verb Object
TRI	technology reference identification
TRI&A	technology reference identification and analysis
TSOTAB	Technology State of the Art browser
UIMA	Unstructured Information Management
URED	Unified Research and Engineering Database
URI	Uniform Resource Indicator
URL	Uniform Resource Locator

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From – To)
05-12-15	Final	

4. TITLE AND SUBTITLE

Proof of Concept Assessment for the Use of Natural Language Processing to Maintain and Update the DoD Technologies Knowledge Base (DTKB)

5a. CONTRACT NUMBER

HQ0034-14-D-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBERS

6. AUTHOR(S)

Steven P. Wartik, Francisco L. Loaiza-Lemos, Anna Vasilyeva, Thi U Tran

5d. PROJECT NUMBER

AK-5-3782

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES

Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER

D-5685
H 15-001227

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Glen J. Stettler
Director, ASD(R&E)/RD/Technology Security Office
Research Directorate
Office of the Assistant Secretary
Research and Engineering (OUSD(AT&L)
2001 N. Beauregard Street
Alexandria VA 22311

10. SPONSOR'S / MONITOR'S ACRONYM

ASD(R&E)/RD

11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Project Leader: Robert Rolfe

14. ABSTRACT

In 2013 the sponsor asked the Institute for Defense Analyses (IDA) to conduct a study regarding possible courses of action in order to answer an inquiry into the status of the MCTL activities by the Office of the Inspector General (OIG). Among the various COAs explored IDA recommended: to re-invent, in the near term, the MCTL as a shared and integrated set of existing information sources — thereby creating a common, dynamic, classified, proprietary, DoD technologies knowledge base (DTKB) reflecting technology velocity, trajectory and disruptive changes to support stakeholders, communities of interest, and other SMEs. In 2014 the sponsor provided initial funding to explore possible implementations of the DTKB concept. This report describes the results obtained pertaining to the use of natural language processing (NLP) technologies to maintain and update a future DTKB.

15. SUBJECT TERMS

Natural Language Processing (NLP), Militarily Critical Technologies List (MCTL), DoD technologies knowledge base (DTKB), Technology Reference Identification (TRI), structured summarization, Natural Language Generation (NLG), General Architecture for Text Engineering (GATE), entity relation extraction (ERE), technology taxonomy, critical parameters, technology ontology, Web Ontology Language (OWL).

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Glen J. Stettler
Unclassified	Unclassified	Unclassified	Unlimited	60	19b. TELEPHONE NUMBER (Include Area Code) 571-372-6443