ABSTRACT

Cybersecurity systems are comprised of a wide variety of interconnected systems. If these individual systems are not sufficiently designed and implemented to work well when integrated, additional opportunities for adversaries to hide and attack are created. To better enable researchers to broaden their scope and work together, they must be able to explore research in other areas and work across disciplines to create standards for communication and systems that work synergistically. This paper describes Department of Homeland Security (DHS) Assistant for Research and Development Tracking and Technology Transition (DART3), a web-based semantic tool and repository designed to capture US federally funded research and development (R&D) project descriptions and R&D requirements to facilitate technology transitions.

1. INTRODUCTION

Cybersecurity systems are comprised of a wide variety of interconnected systems. If these individual systems are not sufficiently designed and implemented to work well when integrated, additional opportunities for adversaries to hide and attack are created. To better enable researchers to broaden their scope and work together, they must be able to explore research in other areas and work across disciplines to create standards for communication and systems that work synergistically.

Department of Homeland Security (DHS) Assistant for Research and development Tracking and Technology Transition (DART3) is a web-based semantic tool and repository designed to capture US federally funded research and development (R&D) project descriptions and R&D requirements to facilitate technology transitions. It is currently being piloted by two government agencies and plans are in place to expand its use to remote repositories and to create a classified version.

Before the development of DART3, the process of matching a requirement to a funded project was very labor intensive. Many funding agencies (e.g., National Science Foundation (NSF), Defense Technical Information Center (DTIC)) store their project data in a relational database, and each database may have a unique schema. For instance, in some project databases the project is described by an abstract, while in another project database the project is described by a description. This means that an individual researching the relationship between projects and requirements needs to understand the different vocabularies used by different agencies. Furthermore, there is much repetitive work in searching for projects across different agencies’ project database interfaces, if such an interface even exists. In addition, the person completing the requirement/project matching must have knowledge of the existence of all such databases. This is an extremely time and labor intensive process; many projects may not even be discovered. The analyst who is exploring the requirement/project relationships may also be biased toward a particular project if, for instance, he or she has recently attended a presentation about the project. For these reasons, an automated algorithm for requirement/project matching simplifies and streamlines the approach and supports more objectivity in the selection process.

DART3 was originally designed to identify R&D projects that meet R&D requirements for operational systems and to use the information to coordinate with the principal investigators (PIs) of the R&D in order to assure the results meet the needs of DHS’s Office of Cybersecurity and Communications (CS&C). It has been expanded to use in organizations outside of CS&C. DART3 helps generate and communicate a set of transition activities to be undertaken to accomplish the technology transition. DART3 can also be used to identify CS&C requirements that do not match any research descriptions. These represent gaps that could be addressed in future research.

Over time, other federal agencies and organizations have expressed interest in using DART3 to match R&D projects against their own requirements. In addition, some researchers have utilized the system as a tool to connect with other researchers interested in similar or complimentary domains. These various uses, both within DHS and across agencies, make DART3 a valuable connection and collaboration tool for researchers interested in advancing technology in holistic cyber security in the ecosystem.

Section 2 will provide a DART3 system overview; section 3 describes the DART3 Lucene matcher and interface and section 4 will describe DART3 in actual use. Section 5 discusses the organization of requirements and section 6 details the technologies behind DART3. We conclude in sections 7 and 8 with future work and a final summary.
(CS&C), but other agencies’ requirements could easily be added. A classified version of DART3 is also under consideration.

By exploring projects and requirements, matches may be discovered for technology transition. DART3 runs a matching algorithm on a regular basis to automatically suggest matches between projects and requirements. This fuzzy matching uses a Lucene algorithm [1] and utilizes the keywords of requirements and the textual descriptions of projects. Details of the matching algorithm are discussed in Section 3.

The DART3 Projects page displays an alphabetical list of all projects available for transition. A similar page is available for listing all requirements. New projects and requirements can be automatically imported into the datastore and run as new input to the matching algorithm.

By clicking on a project name from the list, details of the project are displayed, as shown in Figure 1. The semantic information available on the page makes it easy to browse through projects with related information, for example, other projects with the same performer or organization, more on the semantic relationships in the database are discussed in Section 6. Project data contains PI, performing organization, contact information, dates, project description, keywords, and funding. The page also displays a list of Suggested Requirements; this is the ranked list of matches as output from the fuzzy matching algorithm.

Alternatively, when browsing through requirements, the Requirements page displays the descriptions of the requirement, the associated keywords, the requirement area, and Suggested Projects to meet this Requirement. A Requirement page is shown in Figure 2. When viewing the suggested projects to meet a requirement, the user may explore the project descriptions by clicking on the project name. Note that browsing and transitions can begin either from the project or the requirement perspective.

Moving between projects and requirements, exploring potential matches, and viewing detailed semantic information about each is simplified through the DART3 semantic wiki interface [6]. For the requirement shown in Figure 2, links are displayed to the top five projects suggested that partially meet the requirement, as determined by the DART3 matcher. These project matches can be explored to determine if they really do match and make sense for transition. Occasionally, two or more projects must be combined to meet a requirement. To accomplish this, DART3 allows Transition Groupings.

Project pages also contain a Transition This! button, as seen on the bottom of Figure 1, which can be clicked to launch an assistant to guide the transition process and produce a transition activity and tracking sheet.

DART3 provides assistance with selecting and tracking transition activities once a match and a transition have been chosen. This process is based on the Transition Planning and Assessment Model (TPAM) [2]. Software prototypes consist of several somewhat independent elements. These elements can be evaluated independently and documented separately. The main elements defined by [2] are Architecture, Code, Design, Requirements, Algorithm, User Interface, Code, Concept, and Data. Several of these are available for use in DART3 and are shown in Figure 3. Additional elements can be defined. DART3 tooltips provide assistance in choosing elements. When the user hovers the mouse over each element, a description is shown. Typically, two to three elements are selected for transition. In Figure 3, the Design element is selected.

Once an element is chosen, the characteristics of the element are displayed. Characteristics are pertinent attributes of an element that can be assessed. Typically four to six characteristics of each selected element are chosen. In Figure 3, the Security characteristic is selected. This means that, of the available characteristics of the design of the R&D project, the security design is desired for transition into the operational system.
After a characteristic is selected, the conditions of the characteristic are shown. Each characteristic, upon assessment, will assume one of several conditions. Only one condition per characteristic can be selected, and they will determine the transition activities list. In Figure 3, **Incompatible mechanisms** is selected, meaning that the design security mechanisms of the source R&D project and target operational systems are incompatible.

Once all of the elements, characteristics and conditions have been selected and saved, a summary of all required transition activities is displayed in a list at the bottom of the page, as seen in Figure 3. The user can then explore this list in order to formulate the transition proposal portfolio that makes sense for this project—requirement match. The transition activities list is an editable tracking sheet for recording transition activity status, duration, cost, staff, staff roles and any critical notes. Note that activities can be added, edited, and deleted from this tracker. Reporting mechanisms are also provided by DART3 to show timelines and schedules. The transition page may also be exported as a PDF file and printed.

The automatic matching is completed by doing a web-like search over project descriptions, where the query is made up of the requirement keywords. The query is expanded through use of a fuzzy query and synonym expansion. A fuzzy query is a query in which the correct search is made regardless of minor misspellings of the query terms. Synonym expansion means that not only is a requirement keyword searched for, but all of its synonyms, as well. Synonym expansion is particularly useful for DART3, as different users may apply different keywords to requirements or projects that have similar meaning.

The automatic matching algorithm is performed in a program outside of the wiki environment. A SPARQL query is performed which returns all of the project and requirement information for input into the matching application. Apache Lucene is a text search engine library, written in Java, that was used to implement the matching algorithm [1]. An index of all project descriptions, titles, and keywords is created. A fuzzy query with synonym expansion is completed using the requirement keywords, and the 20 projects with the highest scores may be used to complete the Suggested Projects to Partially Meet This Requirement and Suggested Requirements sections of the requirement and project pages, respectively. The synonyms are found using WordNet [8]. The matches are then written back into the wiki using SPARQL. As the project and requirement information may change, the automatic matcher is run on a regular basis to update the matches.

It is important to note that the automatic matches are suggestions. The matches are made based on the scoring algorithm of Lucene, which is a text similarity algorithm. It is possible that a project description may include requirement keywords for a requirement that it does not meet. It has been observed, however, that in most cases the top several matches for a requirement have been reasonable based on evaluation by subject matter experts (SMEs).

In DART3, the user only has to browse the Requirement or Project pages in order to see these matches. The twenty best matches are made available, with manual matches at the top of the list. Manual matches are described in the next section.

3. **MATCHING**

3.1 **DART3 Automatic Matching**

As described in Section 2, DART3 provides the capability to automatically match DHS requirements to funded R&D projects. All project information and DHS requirements have been exported from their original databases, and imported into DART3. With all of the project and requirement data stored in DART3 in a centralized and uniform manner, a matching algorithm can be run.

The automatic matching is completed by doing a web-like search over project descriptions, where the query is made up of the requirement keywords. The query is expanded through use of a fuzzy query and synonym expansion. A fuzzy query is a query in which the correct search is made regardless of minor misspellings of the query terms. Synonym expansion means that not only is a requirement keyword searched for, but all of its synonyms, as well. Synonym expansion is particularly useful for DART3, as different users may apply different keywords to requirements or projects that have similar meaning.

The automatic matching algorithm is performed in a program outside of the wiki environment. A SPARQL query is performed which returns all of the project and requirement information for input into the matching application. Apache Lucene is a text search engine library, written in Java, that was used to implement the matching algorithm [1]. An index of all project descriptions, titles, and keywords is created. A fuzzy query with synonym expansion is completed using the requirement keywords, and the 20 projects with the highest scores may be used to complete the Suggested Projects to Partially Meet This Requirement and Suggested Requirements sections of the requirement and project pages, respectively. The synonyms are found using WordNet [8]. The matches are then written back into the wiki using SPARQL. As the project and requirement information may change, the automatic matcher is run on a regular basis to update the matches.

It is important to note that the automatic matches are suggestions. The matches are made based on the scoring algorithm of Lucene, which is a text similarity algorithm. It is possible that a project description may include requirement keywords for a requirement that it does not meet. It has been observed, however, that in most cases the top several matches for a requirement have been reasonable based on evaluation by subject matter experts (SMEs).

In DART3, the user only has to browse the Requirement or Project pages in order to see these matches. The twenty best matches are made available, with manual matches at the top of the list. Manual matches are described in the next section.

3.2 **DART3 Manual Matching**

There may be some cases for which a DART3 user knows of a particular project that matches a requirement. The user may not want to rely on the automatic matcher to create this match, or it may be a match that the automatic matcher would have trouble discovering. Having all of the projects and requirements in one place provides a good opportunity for insight that a particular project matches a requirement, even if the description does not match with requirement keywords. DART3 allows the user to create such a match from a Project page.

Manual matches will always appear at the top of the list of matches for a given project or requirement. Furthermore, if the project or requirement description changes, the manual match remains, but has its `outdated` flag set to `true` the next time the automatic matching algorithm is run. The user may then determine if the manual match is still valid and edit the match so that `outdated` is set to `false`. If the match is no longer valid, the user can delete it. At the bottom of a project page, as shown in Figure 1, a dropdown of all available requirements can be used to make a manual match for a project.

4. **DART3 IN USE**

The first input to DART3 requires the determination and elucidation of the needs of the operational areas, or targets. This step is critical because, without proper specification of the requirements, the matching process will not locate good project matches.
The next step is to discover ongoing federally funded research projects, whether they are funded by DHS Directorate for Science and Technology (S&T), or by another entity within the U.S. Government (e.g., Defense Advanced Research Projects Agency [DARPA], United States Air Force [USAF], Department of Energy [DOE], etc.). Universities or other research organizations may perform the research. By attending conferences and communicating with other agencies, potential projects can be identified. Once an accurate, concise summary of a project is written, it is imported into DART3. Projects can be inserted one at a time or in bulk insert.

As these two processes feed validated needs and projects into DART3, the tool matches needs with projects that will potentially satisfy the needs, in whole or in part. Once DART3 provides matches, transitions can be initiated.

5. REQUIREMENTS
For the first (and current) release of DART3, 92 requirements, broken into 19 technical topic areas, were imported into the DART3 repository
Each technical topic area has an abbreviation and title. They include:

- **SWA** Software Assurance
- **MTC** Security Metrics
- **USE** Usable Security
- **INS** Insider Threat
- **NET** Secure, Resilient Systems and Networks
- **MAL** Malware Analysis
- **NMM** Network Mapping and Measurement
- **IRC** Incident Response Communities
- **CBE** Cyber Economics
- **DPV** Digital Provenance
- **NIC** Nature-Inspired Cyber Health
- **INF** Information Sharing
- **SIT** Situational Awareness
- **SUP** Supply Chain
- **RSP** Incident Response
- **SCL** Scalability
- **ADV** Adversary Investigation
- **COM** Communications
- **IDT** Identity Management

Each requirement is then assigned to a category and assigned a number and a title, for example **SWA-1**, in the *Software Assurance* topic area:

**SWA-1 Cloud Security** - Improve security for cloud computing where operations are heavily reliant on personal electronic devices (PEDs) and other previously non-web-enabled devices in an environment where operations and infrastructure are disconnected yet globally interconnected.

As other agencies with differing requirements have expressed interest in the DART3 system, it may be advantageous to express requirements so they can be easily searched and disambiguated by including another layer of identification. For example:

<table>
<thead>
<tr>
<th>Source</th>
<th>Category</th>
<th>Number</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org1</td>
<td>SWA</td>
<td>1</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>Org1</td>
<td>NET</td>
<td>3</td>
<td>Advanced Recovery Architectures</td>
</tr>
</tbody>
</table>

This would enable clear distinctions, flexibility and advanced semantic search capability.

6. TECHNOLOGY
The semantic wiki technology DART3 is built upon provides many advantages for DART3 users. When data is imported into DART3, links in the data are turned into semantic relationships. These semantic relationships are the enabling technology behind the ease of creating new links between projects, requirements, transitions, keywords, individuals, and organizations. DART3 also provides a common interface to anyone familiar with Wikipedia or other Mediawiki-based wikis [7].

6.1 Ontologies vs. Taxonomies
Ontologies are the structural frameworks for organizing semantic information. They provide unique benefits in discovery, flexible access, and information integration due to their inherent connectedness; that is, their ability to represent semantic, conceptual relationships [4]. This connectedness and ability to model coherent, linked relationships is a distinguishing characteristic of ontologies as compared to conventional hierarchical structures, such as taxonomies. Figure 4 shows a traditional, hierarchical taxonomy and Figure 5 depicts a connected ontology. DART3’s database resembles a graph, much like the ontology depicted below.

![Figure 4: Taxonomy](image-url)
6.2 Searching DART3’s Data
A user may take advantage of the connectedness of the data in DART3 in order to search through its data in several ways.

First, the user can search for any information throughout the system using a basic web-like search provided by Mediawiki. For example, the result of a search for “security” is the page shown in Figure 6.

Second, the user can search using the semantic relationships in the data, which are modeled in DART3’s ontology. These relationships are made into browsable links on the wiki pages by Semantic Mediawiki, providing users with the ability to traverse the database by browsing DART3’s pages. For example, the same Security keyword page shown in Figure 6 could be reached by clicking on the Security link under a project or requirement’s keyword list. The projects and requirements that contain Security as a keyword are displayed on the Security keyword page. A user could then click any of those links to reach other projects or requirements tagged as related to security.

Finally, these semantic relationships can also be traversed through queries created through Semantic Mediawiki. Standing queries are used to populate the dynamic lists as shown in Figure 6. In this example, every keyword page is exactly the same, except that the page’s title, in this case the keyword Security, is passed into the query as a parameter. This allows each keyword page to dynamically display the current projects and requirements associated with it. Since these queries are dynamically loaded when the page refreshes, the lists are always up to date with the database.

6.3 Advantages of Semantic Relationships
These relationships could be stored as rows in a conventional relational database, such as MySQL. They could be made to be parsed into the same links and queries discussed previously. Mediawiki stores and generates its own page links created by users with the “[[page]]” syntax. However, there are many advantages that using a semantic database gives DART3.

First, a semantic ontology allows both users and machines the ability to read the data equally easily without any additional computation to parse the database. Since the data is stored semantically, a machine reading the data has full context of how a single piece of information fits into the graph of knowledge stored in the database.

Second, the data is stored, not as a series of tables that must be indexed and joined, but as a singular graph. A good illustration of this can be seen between Figure 4 and Figure 5. Any data for a single entity, such as a project, requirement, keyword, or performing individual, is referenced by a single node in the graph. All data pertaining to that entity form edges in the graph, linking the entity they represent with other entities. This explicit representation of the relationships between data is what gives semantic databases their power.

The final benefit of modeling data in this way is that new data fields can be added ad-hoc. This means that DART3 can evolve its ontology for projects or requirements over time without having to undergo major database changes. These changes to the ontology can be picked up by machines as easily as humans, allowing for easier automation.

6.4 Using and Evolving DART3’s Ontology
A user may exercise these capabilities to update DART3’s ontology and data for a page by editing it. When a user edits a page in DART3, the ontology and database are updated automatically. Fields or links added to a page generate new semantic relationships in the database. These relationships are then available for other users to use for browsing and querying. For example, users could create their own property for a project, called Related websites. This updates the ontology, allowing users to then add Related websites to any project page. If any two projects had the same Related websites, then a link between them could be established, further enhancing the richness and connectedness of the data.

The matching algorithm described in Section 3 makes use of this power, in the same fashion, to more easily retrieve, reason over, and create relationships in the data. As new relationships are added to the ontology that could enhance the matching algorithm’s effectiveness and since the algorithm already has the rest of the context already for the new data, the matcher could be modified with minimal effort to include these new relationships in its calculations. It could also be used to create other, new relationships in the data, just like a user, which could be used for even further browsing, queries, and reasoning.

7. FUTURE WORK
An additional capability that would be useful for the matcher is an algorithm that uses semantic equivalence to discover matches. For instance, determining that trustworthy computing and survivable systems refer to similar ideas would result in an automatic match being made. Currently, this capability could be
hard-coded if such equivalences were pre-determined. However, a more powerful tool should be considered, such as, something similar to WordNet for phrases instead of words.

Standardizing the methods used to describe R&D projects and requirements would significantly improve both the usefulness of this tool and the sharing of R&D information in general. The standardization of these descriptions would promote the use of shared database/web services that could be easily aggregated and searched. For DART3, this would provide the ability to incorporate the changing landscape of R&D projects automatically, providing updated transition matches and helping to further automate the identification of R&D that meets critical operational requirements. If R&D requirements were also standardized, then both ends of this process could be easily automated, making the system available to a much broader range of United States Government and civilian personnel. RSI will be considering approaches along these lines to assist both DHS and other departments and agencies in the identification and transition of technology that meets critical operational needs.

8. CONCLUSION
The DART3 tool enables researchers to connect with other researchers in their field of interest and to reach across disciplines so that systems can be designed to work together. This advances the goal of creating a holistic cyber security system.

9. REFERENCES