KMAPPER – AN AUTOMATED KNOWLEDGE ASSETS DISCOVERY APPLICATION IN SUPPORT OF THE ARMED FORCES

Régine Lecocq*, Alexandre Bergeron Guyard, Marc-André Morin
DRDC Valcartier
2459 Pie-XI Blvd North
Quebec City, QC, G3J 1X5 Canada

ABSTRACT

This paper presents a newly developed knowledge mapping (k-mapping) application called *"KMapper"* along with its underlying multidimensional approach. The KMapper, as a network science technology, is an automated application allowing the discovery, identification, localization, access and support for the exploitation of KAs by the Commander and the Soldier. Subsequently to presenting the concept of k-mapping in general, we describe the foundations for the KMapper developed by DRDC Valcartier. We then discuss the approach and how knowledge is structured around 4 dimensions that are organised in a KMapper core ontology. We then go into greater detail about the KMapper Alpha prototype, describing how every task leading to knowledge discovery and knowledge mapping is implemented. Along with the preliminary and promising results from the KMapper Alpha prototype application, we also illustrate the pragmatic challenges that were met and discuss those that are to be addressed in future versions in order to better support the Armed Forces.

1. INTRODUCTION

The Canadian Forces are facing, to a greater extent then ever, challenging operating environments (DRDC, 2006). Our Armed Forces have to operate in environments characterized by uncertainty, instability and risk. Moreover, the security challenges being faced will not stay confined to the external arena. "In an increasingly interconnected, interdependent and information-based world, lines between the external and the domestic will be increasingly blurred" (DND/CF, 2007). Therefore, this will require "forces that are combat-effective, but also highly mobile, adaptive, networked, sustainable and capable of operating in a Joint, Interagency, Multinational and Public (JIMP) context" (DND/CF, 2007). Thus, in domestic operations or abroad, the diversity of missions increases. In this age of information and knowledge, the technological complexity being faced is intensified by the intricacy of the different military and non-military organizations that the commander has to compose with, and which are an intrinsic part of the situation. These stakeholder organizations hold critical pieces of knowledge assets (KAs) for mission success. These KAs are considered crucial by the militaries to first fully understand the situation at hand to then make effective and accurate decisions. Unfortunately, as the number of involved organization increases, it is also incontrovertibly more difficult to identify, what organization holds which critical KA. Therefore in order for the military personnel to adequately exploit those situational KAs, these ones need first to be identified, located and subsequently made available.

2. K-MAPPING AND THE KMAPPER CONCEPTS

The term "knowledge mapping" (k-mapping) has emerged from the growing field of knowledge management, but its foundations can be traced in different fields. The first element noticeable about k-mapping is that researchers or practitioners refer to it indifferently as a term, a methodology, or an approach. For the "knowledge mapping" as a specific term, a closer look (Lecocq, 2006) reveals the usage of different meanings for it as for instance: "knowledge audit" (NELH, 2008), "concept mapping" (Trochim, 2002), or "knowledge modelling" (Schreiber, 2002). K-mapping can also be perceived as responding to one or the other of the three following approaches: the conceptual, the procedural (Kang, 2003), and the social one (Cross, 2002). However, usually, each of these approaches is only encountered from its single perspective. The novelty of this research first resides in the fact that we combine the value of each one of those three approaches into a single one called the *"Multidimensional K-mapping Balanced Approach"*. Then, we also add the value of a fourth approach named *"Knowledge Artefacts"* (K-Artefacts). Such a k-mapping approach in the defence domain, aims at first enhancing individual and collective understanding of a situation being faced; secondly facilitating the sharing of such an understanding through a commonly shared context; and finally increasing collaboration opportunities for the purpose of mission success. During the last year, most of the research effort reported here has focused on the development of a k-mapping application, the KMapper corresponding to such a conceptualization.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 DEC 2008	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
KMAPPER An Automated Knowledge Assets Discovery Application In Support Of The Armed Forces	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **DRDC Valcartier 2459 Pie-XI Blvd North Quebec City, QC, G3J 1X5 Canada**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002187. Proceedings of the Army Science Conference (26th) Held in Orlando, Florida on 1-4 December 2008, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

2.1 A Single Balanced Approach composed of Four Dimensions

The KMapper is organized around a single multidimensional approach integrating the value of four different standpoints, which are named, *dimensions*: the Social, K-artefact, Conceptual and Process dimensions.

2.1.1 Two Dimensions for KAs Categorization

The Social and the K-artefacts dimensions are two types of categories under which KAs are being gathered and organized within the KMapper. For these dimensions, and through different technological and social networks means, the KMapper extracts KAs mostly automatically and then stores and creates relevant links between them and meaningful concepts. The creation of these links is also a means to discover knowledge using the KMapper. For instance, e-mails between individuals on the topic of "chemical spills" can be linked to papers written on the same topics and subsequently be of usage to an individual having to locate expertise in the domain within the context of a related situation. The KAs being organized under the Social dimension are sources of knowledge that can be: experienced or knowledgeable individuals, specialized groups, or else organizations working in specific domain areas. The K-artefacts dimension also aims at organizing discovered KAs. However, in this case, it refers to sets of KAs that can be considered as explicit knowledge such as documents, databases or websites for instances.

2.1.2 Two Dimensions as Presentation Axes for KAs

Once the KAs are discovered and organized under the Social and K-artefacts dimensions, they are visually presented to the user around the two other dimensions: the Concept and the Process dimensions. Presenting the KAs along those two dimensions permits the users not only to comprehend the KAs within a meaningful context but also to provide new knowledge to the user by making apparent unexpected links between KAs and new concepts or specific process stages. Indeed, the concept dimension of the KMapper permits a visual presentation of a specific domain ontology to the military user with its related concepts and the relations existing between them. These act as contextual pieces to position the identified KAs from the Social and the K-artefacts dimensions. By doing so, the end-users can immediately ascertain to which concepts a KA is related. Similarly, the process dimension covers the key processes being worked with by the targeted group of users of the KMapper. Whenever a specific stage of the process is being worked with, the KAs that should be prioritized in the context of that specific process are presented on the KMap. Here again, the process dimension acts as a contextual element permitting the positioning of the KAs.

2.2 Ontology-Based Application

The KMapper application is an ontology-based system. The application requires several ontologies in support of different functions. These functions can be to support the search capability in order to retrieve information about KAs relevant to the end-user; convey a significant context to visually display the identified KAs; or else to provide the military end-users with a certain level of shared context for common actions. The ontologies also respond to the need for knowledge inference with k-mapping and it contributes to the *"Knowledge Inference"* service.

2.2.1 The KMapper Core Ontology

The "KMapper core ontology" supports the application itself. It permits the definition and description of concepts and their relationships related to the KMapper application, as well as its structure, dimensions, etc. The left side of the figure 1 offers different available KMapper core ontology classes; these classes provide information about the dimensions and sub-elements of the dimensions to which a KA may belong. The right side of the figure 1 provides relations that can be applied to the specific selected class from the left side. Finally, for each KA, once all elements identified, the information can be stored in the knowledge base for further exploitation by the KMapper application.

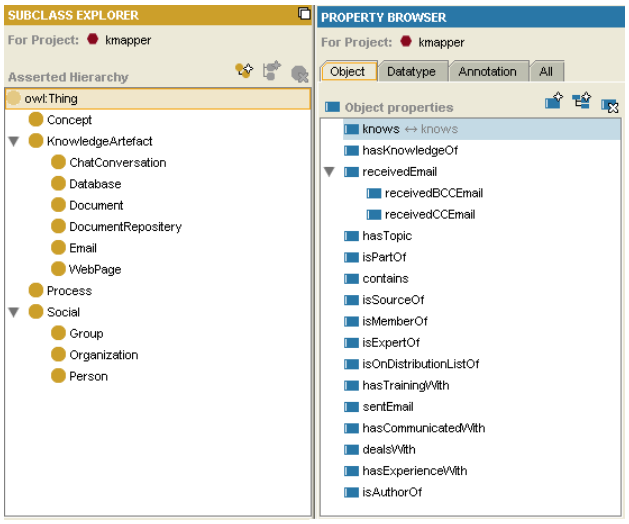

Fig. 1: KMapper core ontology classes and relations

2.2.2 The Domain Ontology

The domain ontology is considered as the backbone of the KMapper, as it supports the search engines in attempts to retrieve information about KAs of

significance to the end-user. In support of this knowledge extraction capacity, the application searches new data sources to extract KAs but it also benefits from data sources identified a priori and injected in it to extract additional KAs. It is around the ontology structure that the extracted and injected KAs are then organized and linked to one another to be visualized. Finally, the domain ontology provides a certain level of shared context for common action between military end-users.

2.3 Knowledge Assets vs. Knowledge

While the KMapper points to KAs pertaining to specific domains as opposed to presenting the knowledge itself, whenever possible it also provides the user with an access as direct as possible to these KAs. This is of particular interest as it permits a dynamic exploitation the KMapper. Indeed, instead of having to perpetually refresh or renew the actual knowledge, the data sources themselves are somewhat more stable in time.

2.3.1 Automated KA Discovery and Manual Input

The KMapper combines the value of automated extraction as described by Ehrlich (Ehrlich, 2003) and manual inputs done by individuals who have the knowledge of their own environment and context. Each individual, as a source of knowledge, knows other groups, organizations, subjects, documents, etc. meaning other KAs relevant to a concept from the domain ontology. Such a specific knowledge should not be overseen but these manual additions to the KMap (and indirectly into the database) are supplemented by updates where the majority of the KAs are discovered via the automated information gatherer services as described in Section 3.2.2.

2.4 Added Value of Knowledge Mapping

Knowledge mapping as exposed here brings its first added value through the identification and localization of KAs for further exploitation by the end-users. This output can be pursued to reach desired outcomes as for instance an increased collaboration within or between organizations, a reduced time-to-competency, enhanced knowledge awareness, or else a higher level of understanding of the situation being faced.

2.4.1 Making Sense of the Situation

At the offset of a mission, as well as while the mission unfolds; knowledge mapping can support the individuals in their efforts to understand a situation as quickly as possible. As the concept dimension is based on domain ontologies, by exploiting it the individual develops an understanding of the context of the situation and its significant concepts. Also, through this net of concepts as well as related KAs, the user reaches another type of understanding being the one of the social instances potentially involved with the situation.

2.4.2 Building Organizational/Group Memory

As mentioned, even if numerous KAs are automatically identified and located, each of the users is himself/herself a KA and should therefore be able to add his/her own knowledge to the knowledge base. This important feature is key in the military domain where rotations are frequent and the need to build collective or mission specific memory is even more essential than in other types of organizations. This given, new individuals arriving can benefit from knowledge previously developed. Similarly, for individuals working in the same group at a given time, the KMapper also provides them with the ability to share comments, workspaces as well as specific discovered KAs.

2.4.3 Increasing Collaboration

Highlighting specific KAs related to key concepts also, in itself, brings an additional piece of understanding about existing collaborations and the situation. For instance, in the situation where a pandemic outbreak of a disease occurs in a foreign country, discovering that a specific branch of the national department of foreign Affairs is a source of knowledge (along with the fact that it holds certain roles and responsibilities) can be important to develop collaborative behaviour with its people. Currently, collaborative actions between Forces and other friendly instances are perceived as positive behaviours and clearly key to mission success. In order to perform effective collaborative behaviours; it is required to understand its pursued benefits as well as its drawbacks to avoid. Researchers (Lecocq et al., 2007) have identified that some of the key drawbacks of collaboration are its time consuming aspect and the fact that it sometimes requires to collaborate with too many instances. Indeed, effective collaboration takes time and collaborating with too many instances widespread collaborative results leaving the military with disappointing experiences. By mapping the groups/organizations and their specific knowledge in a domain it permits to have focus collaborative efforts with critical instances.

2.4.4 Identifying Critical but Restricted KAs

In some cases, specific identified KAs can be classified and therefore not even known by the user depending on the user's own security level. In other cases, the general content of these KAs can be known whereas the details cannot be accessed. Therefore, only

some classes from the domain ontology, if pertinent, will be matched and moreover, the KAs will not be directly accessible from the KMapper. One of the added values of the KMapper is its capacity in this latest case to provide the user with some of the metadata that will permit an indirect access to these KAs, for further exploitation. An instance of this can be metadata about the name, phone number, and person of contact from the organization in charge of a specific key database that is of restricted access. Furthermore, this capability of the KMapper has been identified as a potential catalysis to efforts from different government departments around key KAs that should require more sharing.

3. THE KMAPPER ALPHA PROTOTYPE

The KMapper aims at: visualizing KAs, related concepts and their links based on dimensions; searching for expertise held by individuals/groups and getting their contact information; linking the different KAs to the military process being used; identifying and locating relevant k-artefacts like documents or databases entries, as well as exploiting the mechanisms for accessing the data sources containing them. The following sections detail how the prototype addresses these various aspects. We explain how the KMapper is built, how it works, which challenges were encountered, and how some were solved.

3.1 Alpha Prototype Architecture

The KMapper Alpha prototype must have the flexibility to access different data sources, implement different knowledge treatment capabilities, and respond to a wide variety of users' needs.

3.1.1 Service-Oriented Architecture

A Service-Oriented Architecture (SOA) is a loosely coupled software architecture which aims at translating business processes into Web services. A Web service is a "URL-addressable software resource that performs functions and provides answers" (Seybold, 2002). Numerous factors have motivated our choice for this architectural approach. Web services allow different type of interfaces to remotely access distributed data sources and applications through a network. This facilitates the acquisition of data from different locations. A Web Service can have a dedicated client application, but it can also be accessed through Web browsers, wireless devices, agents or other Web Services. This gives great flexibility for client development.

3.2 Alpha Prototype Services

We will discuss every application service working in chronological order. We will start from the first services required to get the KMapper running, and progress along the different components used by the system. Figure 2 illustrates this chronological process, starting (on the left) with *administration* services and ending (on the right) with *visualization* services and search capabilities.

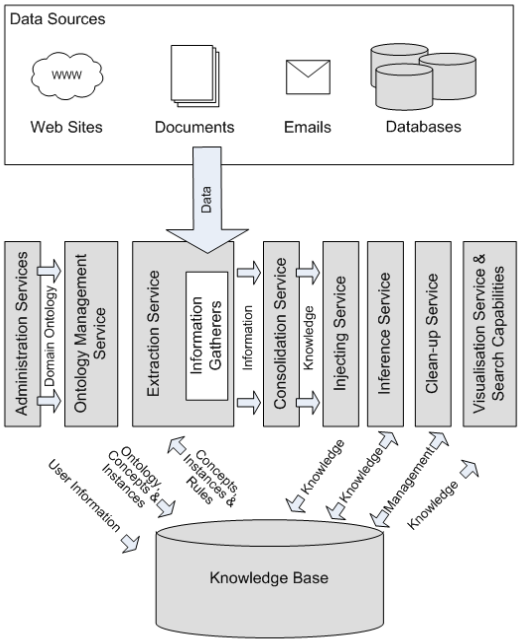

Fig. 2: KMapper application services

3.2.1 Ontology Management and Administration Services

The very first thing that is required for the KMapper to work is a domain ontology, which is used to identify the concepts one is interested to look for in the various data sources. Through the administration module, a knowledge engineer can load an ontology into the *ontology management* service (OMS). This service provides access to the ontology taxonomy (key concepts), as well as to the rules relative to it. The OMS is also where instances of some concepts can already be found (e.g.: Osama Ben Laden as an instance of a terrorist). The *administrative* service is used to manage various users and their roles. In version Alpha, the regular user can access the data and visualize it according to his/her preferences and the administrative user – the knowledge engineer – can load ontologies in the OMS, modify metadata contained in the Knowledge Base, upload new documents in the system, and manage the users of the system. The main issue with the OMS deals with the fact that not all ontologies have taxonomies (concept hierarchies) that contain concepts labelled in a way that

would be useable as search keys. Sometimes, ontologies will have lists of keywords, provided as fields or comments, but keywords may not always be available. For instance if we consider the P-JC3IEDM ontology (P stands for Protégé) derived from the Joint Command, Control and Consultation Information Exchange Data Model (JC3IEDM) as our domain ontology, we will encounter classes or concept names such as ACTION-OBJECTIVE-AUTHORISING-ORGANISATION. It's highly doubtful that we'll ever be able to extracts a lot of knowledge if we use the concept's name tag as a search key. To solve this problem, it is necessary to provide the knowledge engineer with the capacity to map concepts of the ontology with the proper keywords. The efficiency of this solution will rely strictly on the quality of the concept-keyword mapping that will be provided through the OMS.

3.2.2 Extraction Service

Once the ontology is ready for use, the extraction of key data and information is done through the use of information gatherers composing the *information extraction* service. Information Gatherers are agents capable of connecting to various data sources – Web sources, active directories, databases, exchange servers and document repositories – and retrieving elements of interest. To certain extend, each gatherer is customizable and, for instance, a database gatherer, fetching very specific data from a particular table, will be easily tailored to other databases. A gatherer querying a website's search capability (through a CGI script for instance) will be tailored to the specifics of the particular website (query string and result format) and harder to reuse. The basic principle is that all types of data source are useable by tailoring a specific gatherer for them.

These data sources contain information that pertains both to the Social and K-artefact dimensions. From the moment where the ontology is loaded in the system, a human resources database could be searched in order to find any person that has knowledge, training or expertise on concepts of interest. At the same time, the documents of the repositories can be looked through in order to identify which ones contain concepts present in the domain ontology. This task consists in extracting named entities according to an annotation schema: a data structure containing the domain ontology. It is performed internally by using the free and open source GATE 4.0 (General Architecture for Text Engineering) software (Cunningham et al., 2002). GATE is a development environment for language engineering, and natural text processing. It processes documents, allowing for concept identification and extraction. The website and active directory are used more as "secondary" data sources, intended to add metadata to the social information already gathered (phone numbers, emails, addresses, etc.). Once extracted, the data is forwarded to the *consolidation* service as a potential KA.

The challenge here resides in the customization of gatherers for specific information sources. In the context of SOA systems, such a customization is facilitated by accessing a service bus or discovering data access service. In current life there remain different levels of effort required to query different sources. The problem comes from trying to find specific information in unstructured text. In the case of structured text, it is relatively easy to do so. For instance, it is quite simple to extract information from a DBMS if you have enough understanding of its structure. It is also possible to extract information from the headers of emails, where senders/receivers are structured in a common syntax, separated with specific text markers. But processing unstructured documents (e.g. DOC, PDF, PPT, RTF, TXT, etc.) or semi-structured documents (e.g. HTML, XML, XLS, CSV, DBF, etc.) is problematical. As mentioned, our solution relies on GATE 4.0, a natural language processing (NLP) technique that allows extracting named entities from text by applying programmatic and algorithmic processing resources. Processing resources include summarizers, translators, parsers, and speech recognisers and they typically work from dictionaries, thesauri and grammatical rules. For semi-structured documents, we can benefit from the document structure, such as HTML, where tags are embedded in the page. Therefore, a rule-based task can be performed by GATE in order to extract the author of a website just by retrieving the META tag relative to it. In the case of unstructured documents, JAPE rules can be used. JAPE is a java-based pattern matching language used by GATE. It provides the means to apply a particular grammar to a text in order to extract annotations, or highlights in a document. JAPE rules have to be adapted according to the knowledge domain, so the domain ontology's taxonomy has to be put to use. The rule is applied in conjunction with the parser or the tokenizer, in order to extract in the text lookups pertaining to that specific domain.

3.2.3 Consolidation and Injecting Services

Once information is extracted, the *consolidation* service sorts the accumulated information. This service contains a lot of logic that allows refining information into knowledge, by linking it together, avoiding repetition and contextualizing it. This service is responsible for eliminating identical instances. It is also where some information fusion occurs. If we find three instances of an individual with the same name, the *consolidation* service will consider it as a single person if certain conditions are met (same phone number for instance). Once the accumulated knowledge is consolidated, it is pushed into the database by the *injecting* service. The service makes

sure that the data is properly stored in the database for practical use.

We will expound 2 main emerging issues here; both convey the general sense of what type of challenges may be expected when consolidating data. The first problem relates to identifying multiple similar users as being distinct or not. Consider the case where we have documents A and B, written by John Smith, with no additional information about John Smith available in the documents. Should those two instances of John Smith be considered as the same or as distinct? Sometime a possible solution lies in looking for more information about John Smith. If we trust a data source to contain all possible relevant individuals, and it contains a single John Smith, our problem is solved. However, if no such social data source exists, or if many John Smiths are found, the problem remains. Another complementary solution would be to notify the user or knowledge engineer in order to have him/her proceed to proper verification. The metadata of the different John Smiths could be then edited to clearly identify them as identical or distinct. No matter which solution is considered, this is a limit of the system, dictated by the quality of metadata available, which demands human intervention to be resolved. The second problem deals with being able to identify if two documents are the same. In the Alpha Prototype, if we find two documents with the same name, size, and type (extension), we consider them to be identical. But we have decisions to make when documents share a name, but don't have the same type. Even if the documents' names differ, a user may have produced a word and a .pdf version of the same document and named them differently. As of version Alpha, the KMapper only considers documents that have the same name, size, and type as similar. In later versions, we intend to consider the contents of a document as one of its identifiers. To be able to accomplish this, we will have to go over the document's contents and extract metrics, such as document vector matrixes, that will allow establishing similarity.

3.2.4 Inference Service

This service is essential to discover new relevant knowledge. While all other services have to do with getting what's available and making sense of it, the *inference* service aims at gaining additional knowledge from what has already been gathered. This is done by using well established rules on the knowledge we have accumulated thus far. The *inference* service is somewhat similar to the *consolidation* service in that it also implements complex rules, but this time, in relation with the KMapper ontology. The KMapper ontology contains the concepts for the four dimensions as well as rules that establish how these concepts relate to one another. For instance, a person will be identified as being an expert ("isExpertOf") in a particular field if such information is found in the human resource database. This information could also be inferred if a rule states that a person can be considered an expert of a subject if he or she has written about it and has experience and training on it. In the KMapper Alpha prototype, we have experimented with reasoning using the Semantic Web Rule Language (SWRL) and the Java Expert System Shell (JESS) engine on the OWL-DL KMapper Ontology represented in Protégé.

Numerous challenges arise when attempting to implement such a service. Indeed, some of the inference rules are simple to construct. For instance, we can easily infer that if a person is considered an expert ("isExpertOf") for a given concept, that person can be asserted as having knowledge ("hasKnowledgeOf") of that concept. Obviously, "isExpertOf" is a subset of "hasKnowledgeOf", and can therefore easily be categorized. Things get much harder when we want to start from an element of a superset, and determine if it is also member of the subset. We could also want to determine if being a member of certain collection of sets could determine your membership to another distinct set. For instance, we know that having knowledge of a concept doesn't make you an expert on it. How about having knowledge, having received training on ("hasTrainingWith"), and having written documents on the subject ("isAuthorOf")? This question is complex and yields challenges both from the technological, and the research perspective. The research problems have to do with the construction of the rules. We have to properly evaluate how we want each of our classes and properties to relate to one another. This work is being done in the KMapper ontology, trying to determine necessary and sufficient conditions to assess if an individual can be asserted as a member of a class. It is also done by building separate rules that are executed by a reasoner, in order to extract relevant knowledge. Once that is complete, we need a technological implementation of the rules. As mentioned before, we have experimented with reasoning using the Semantic Web Rule Language (SWRL) and the Java Expert System Shell (JESS) engine on the OWL-DL KMapper Ontology represented in Protégé. In this context, reasoners use three main functionalities to retrieve new knowledge: classification, realization, and rule-based reasoning. A reasoner can classify the taxonomy of the ontology. That means the reasoner will determine how the different classes relate to one another. It will identify if certain classes have an inheritance or equivalence relationship. This becomes useful when we try to determine to which classes an individual belongs. This is the second functionality of the reasoner: realization. Realization is the computation of the exact types of an individual based on the classes it originally belongs to. Finally a reasoner can be used to execute certain types of rules. The SWRL is capable of formulating rules in the form of Horn clauses (disjunction

of literals with at most one positive literal). We end up with rules having a form similar to C1 ^ C2 ^… ^Cn → Cm. Here is a syntax example of SWRL rules:

```
Person(?x) ^ Concept (?y) ^ Document (?z) ^
hasTopic(?z, ?y) → hasKnowledgeOf (?x, ?z)
```

This example would read: "If a person has written a document on a particular concept then this person has knowledge of that concept." This rule can be interpreted by a JESS reasoner. JESS would find all instances of persons having written documents and would add the relation "hasKnowledgeOf" between the persons and document topics. A need that has emerged from the formulation of SWRL rules is the capacity to handle counting in rules. For instance, we want to be able to specify that a person can be considered as an expert of a particular concept if he/she has written four books on it. While counting is not supported by SWRL, Protégé 4 supports cardinality in class descriptions. This brings the possibility to create a subclass of Person called expert, and define it as a person having written at least four books on a concept. The reasoner would then realize all fitting individuals into that new subclass. This looks like a good potential approach at first but it would require the creation of a new class in the ontology for every potential conclusion. Creating new rules would have a direct impact on the KMapper taxonomy, and impact the whole application. To address this problem, a custom rule system, supporting counting, will have to be developed in a future KMapper version. Knowledge inference is one thing that makes the KMapper unique. Being able to fetch useful information from various data sources, extracting knowledge from it and providing an accurate view of the KAs is useful. The real added value comes from new knowledge that can be inferred. Not only are we giving a complete picture of the state of available KAs, we are also moving towards a better understanding of where the information is located, how to access it and who owns it.

3.2.5 Clean-up Service

The *clean-up* service is meant to remove useless or dated entries from the database. If a document is removed from the repository, it must be removed from the database along with all the elements that were related to it only (e.g., author of a single document). Using this service assures that the information displayed on the KMap will be valid and up to date. Removing elements from the knowledge base can impact other elements and relationships. It is therefore capital to execute this task with care.

In the inference service we explained how different KAs can be used by a reasoner to infer new knowledge. This leads to considering the impact, on the knowledge that could have been inferred, of removing particular pieces of knowledge. For instance, let's consider the case where we find a document on terrorism written by John Smith. We could infer that John Smith has knowledge about terrorism. We could add that new knowledge to the database. If, for any reason, that document has to be removed, what would happen of the inferred knowledge? This is a simple example, but we can easily see how more complex cases, spanning over many KAs, and implicating numerous inferences, could come about. It is therefore of the utmost importance to carefully manage deletions from the database in order to avoid gradual corruption of the data. For prototype Alpha, we have kept deletions to a strict minimum, trying to avoid them as much as possible. In future versions, we will have to address this *clean-up* issue. An initial way to do this would be to keep track of all the pieces of information used when inferring new knowledge. Establishing a link between original KAs, the rules they have triggered, and the new information they helped discover. This way, when removing information, we could identify the impact it has on the rules used and conclusions reached using that piece of data. Either a knowledge engineer or a carefully designed automatic process could sort through the remaining knowledge and evaluate its pertinence.

3.2.6 Visualization and Search Capabilities

Once all the knowledge has been processed, the regular user will have access to it through the visualization module. The main functionalities of this module revolve around knowledge visualization and search capabilities. The *visualization* service allows the user to view the knowledge present in the database on a concept-centric map. What this means, is that, to be displayed, any component on the map (social, artefact, or process) will have to be linked to a concept. The user can choose to add a single concept to the map. He can also elect to add a document or a person, in which case both the added element and its related concept will appear. By clicking on an element of the map, the user will be presented with all the metadata relevant to it.

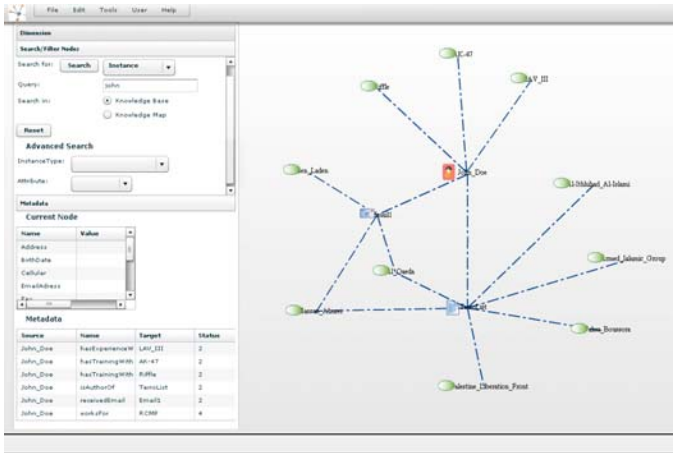

Fig. 3: KMap, metadata, and search capabilities

The visualization service also handles filtering and layout capabilities. Filtering allows the user to view only certain aspects of the KMap (social aspects for instance). The layout manager displays the elements of the KMap in the way that is more suitable to the user. It also handles decluttering, when a lot of elements are displayed. The visualization of the KMapper Alpha uses the Prefuse visualization toolset. Providing the user with a customizable, complete view of the knowledge is always going to be a challenge. Beyond this, there is also a need to help the user notice more important parts of the KMap. This could be attained by giving a visual hint about link importance through thicker lines for instance. The subsequent KMappers will also allow the ranking of relations between KAs. For example, the relation "isExpertOf" between a person and a concept should be considered more important if the person has written every book there is on a subject compare to a single article.

The search capabilities allow the user to look for particular information in the database or the KMap itself. As displayed in Figure 3, having found a relevant KA in the DB, the user can then elect to add it to the KMap. Figure 3 shows a KMap, with different concepts, k-artefacts, and social components. The search fields and metadata are visible on the left hand side of the screen.

CONCLUSION

While the present research is currently delivering significant results within an *"Applied Research Program"* as well as a *"Technology Demonstration Program"* both aiming at increasing situation awareness for the military; k-mapping has many other potential uses for the military. Indeed, some of its other outcomes are foreseen for investigation and trials in some of our key operation theatres. Those other types of value-added can be for instance, to increase the mission memory, collaboration activities or permit the identification of critical but restricted KAs. From a technology standpoint, we have highlighted the challenges faced with the prototype and the various aspects still requiring effort and researches. On top of additional visualization features as well as refining the services, other avenues of research should also be considered regarding the KMapper. For instance, adding a time component to the knowledge would be interesting: allowing for an evaluation about "out of date" pieces of information or else links to historical data. The addition of a geographic feature that would help locate the origin of the KAs on a map would also be useful, especially in the Defence domain. The modification of portions or the whole of the domain ontology and its impact on the application is a key element to be studied within a near future. Finally, a more in depth integration of the "process" dimension is of no doubt one of the most urging requirement for the KMapper, which is currently being worked on.

ACKNOWLEDGMENTS

The authors wish to thank M. Jean Roy and Dr. Alain Auger for their input and support throughout the production of this paper.

REFERENCES

Cross, R. et al., (2002): A bird's eye view: Using social network analysis to improve knowledge creation and sharing, *IBM Institute for K-based Organizations*.

Cunningham, H. et al., 2002: GATE: A Framework and Graphical Development Environment for Robust NLP *Tools and Applications. Proceedings of the ACL'02*

DND/CF, 2007: Land Operations 2021 Adaptive Dispersed Operations, *DND/CF*

DRDC, 2006: Defence S&T Strategy - Science and Technology for a Secure Canada

Ehrlich, K., 2003: Locating Expertise: Design Issues for an Expertise Locator System, *In Sharing Expertise Beyond KM.* Ackerman, M.S. et al. The MIT Press.

Java EE 5 Tools Bundle Documentation and Resources. http://java.sun.com/javaee/sdk/tools/resources.jsp

Kang, I. et al., 2003: A framework for designing a workflow-based knowledge map, *Business Process Management Journal*; Volume 9 No. 3.

Lecocq, R., 2006: Knowledge Mapping: A Conceptual Model, *DRDC-RDDC Valcartier TR2006-118*.

Lecocq, R. et al., 2007: Concepts of knowledge creation, collaboration, and learning for the Canadian military, *DRDC Valcartier TR 2006-138*.

National Electronic Library for Health site. Conducting a Knowledge Audit, http://www.nelh.nhs.uk/knowledge_management/km2/audit_toolkit.asp

P-JC3IEDM. http://vistology.com/onts/onts.html

Schreiber, G. et al., 2002: Knowledge Engineering and Management: The CommonKADS Methodology, *The MIT Press, London*, 455 p.

Seybold, P.A.., 2002: Web Services Guide for Customer-Centric Executives, http://www.psgroup.com

The Prefuse Visualization Toolkit. http://prefuse.org/

Trochim, W.M.K., 2002: Concept Mapping, http://www.socialresearchmethods.net/kb/conmap.htm