7th International
Command and Control Research and Technology Symposium

September 16 - 20, 2002

Québec City, QC, Canada

**********

# FINAL PAPER

**********

**Adel Guitouni**[1], **Anne-Claire Boury-Brisset,**
Defence R&D Canada - Valcartier
2459 Pie-XI North, Val-Belair, QC, G3J 1X5, Canada
adel.guitouni@drdc-rddc.gc.ca, Anne-Claire.Boury-Brisset@drdc-rddc.gc.ca
Phone: +1 (418) 844 4000 ext. 4302, 4392 Fax: +1 (418) 844 4538

**Lamia Belfares, Kabulo Tiliki,**
Department of Operations and Decision Systems,
Faculty of Administration Sciences, University Laval, Québec (Qc), G1K 7P4, Canada

**Nabil Belacel,**
Institute for Information Technology
-e-Business, National Research Council of Canada
2 Garland Ct. (Incutech), Box 69000, Fredericton, NB, E3B 6C2, Canada

**Christian Poirier and Patrice Bilodeau**
Intell@xiom inc.
840, Sainte-Thérèse, bureau 302, Québec (Québec) G1N 1S7, Canada

Possible topics: **Decision Aid Support** or **Information Superiority**

[1] Point of contact

# Automatic Documents Analyzer and Classifier

**Adel Guitouni, Anne-Claire Boury-Brisset,**
Defence R&D Canada - Valcartier
2459 Pie-XI North, Val-Belair, QC, G3J 1X5, Canada
adel.guitouni@drdc-rddc.gc.ca, Anne-Claire.Boury-Brisset@drdc-rddc.gc.ca
Phone: +1 (418) 844 4000 ext. 4302, 4392 Fax: +1 (418) 844 4538

**Lamia Belfares, Kabulo Tiliki,**
Department of Operations and Decision Systems,
Faculty of Administration Sciences, University Laval, Québec (Qc), G1K 7P4, Canada

**Nabil Belacel,**
Institute for Information Technology
-e-Business, National Research Council of Canada
2 Garland Ct. (Incutech), Box 69000, Fredericton, NB, E3B 6C2, Canada

**Christian Poirier and Patrice Bilodeau**
Intell@xiom inc.
840, Sainte-Thérèse, bureau 302, Québec (Québec) G1N 1S7, Canada

## Abstract

Military organizations have to deal with an increasing number of documents coming from different sources and in various formats (paper, fax, e-mail messages, electronic documents). These documents have to be screened, analyzed and categorized in order to interpret their content and gain situation awareness. These documents should be categorized according to their content to enable efficient storage and retrieval. In this context, intelligent techniques and tools should be provided to support this information management process that is currently partly manual. Integrating the recently acquired knowledge in different fields in a system for analyzing, diagnosing, filtering, classifying and clustering documents with a limited human intervention would improve efficiently the quality of information management with reduced human resources. A better categorization and management of information would facilitate correlation of information from different sources, avoid information redundancy, improve access to relevant information, and thus better support decision-making processes. The RDDC-Valcartier's ADAC system (Automatic Documents Analyzer and Classifier) incorporates several techniques and tools for document summarizing and semantic analysis based on ontology of a certain domain (e.g. terrorism), and algorithms of diagnostic, classification and clustering. In this paper, we describe the architecture of the system and the techniques and tools used at each step of the document processing. For the first prototype implementation, the focus has been concentrated on the terrorism domain to develop document corpus and related ontology.

## 1. **Introduction**

In May 1999, the new National Defense Command Centre (NDCC) was commissioned. The mission of the NDCC is to provide a 24/7 secure command and control facility through which the Command staff can plan, mount and direct operations and training activities at the strategic level. Since September 11th, 2001, the level of traffic of messages at the level of the NDCC reached unpredictable peaks. The operators of the Center are overloaded with information that they should handle in real time. The information “digested” by the NDCC represents vital stakes for several other users.

Military organizations and particularly intelligence or command centers have to deal with an increasing number of documents coming from different sources and in various formats (paper, fax, e-mail messages, electronic documents). These documents have to be analyzed in order to interpret their content and gain situation awareness. These documents should be diagnosed and categorized according to their content to enable efficient storage and retrieval. In this context, intelligent techniques and tools should be provided to support this information management process that is currently partly manual.

Automatic, intelligent processing of documents is at the intersection of many fields of research, especially Linguistics and Artificial Intelligence, including natural language processing, pattern recognition, semantic analysis and ontology. Integrating the recently acquired knowledge in these fields in a system for analyzing, diagnosing, filtering, classifying and clustering documents with a limited human intervention would improve efficiently the quality of information management with reduced human resources. A better categorization and management of information would facilitate correlation of information from different sources, avoid information redundancy, improve access to relevant information, and thus better support decision-making processes.

The ADAC system (Automatic Documents Analyzer and Classifier) is developed at RDDC-Valcartier as concept demonstrator and test bed. It incorporates several techniques and tools for document summarizing, semantic analysis based on ontology of a certain domain (e.g. terrorism), and algorithms for automated diagnostic, classification and clustering.

ADAC’s launching agents automatically intercept any new document. Then the document is processed through the following steps (see Figure 1):

1- *Summarization*: provide a synthesized view of the document’s contents;

2- *Statistical and semantic analysis*: index the document by identifying the attributes that best characterize it. Both statistical analysis and semantic processing exploiting domain ontology are carried out at this stage. This produces the document DNA;

3- *Diagnostic*: intercept relevant document matching criteria provided by the user (e.g. document on a particular subject) in order to apply an appropriate action (e.g. alert);

4- *Filtering/classification*: classify/categorize the document in pre-specified hierarchical classes;

5- *Clustering*: assign the document to the most similar group of previously processed documents.

External actions can then be triggered on specific classes of documents (e.g. alerts, visualization, data mining).

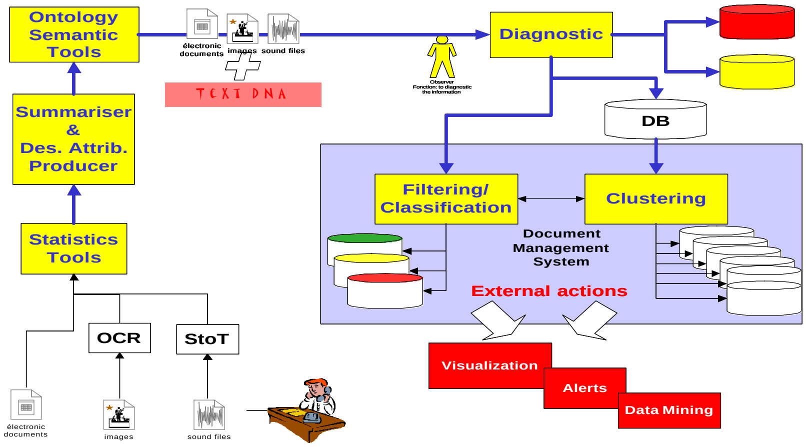


Figure 1: ADAC's Document Processing

In section 2, we describe ADAC system functionalities as well as the approaches and algorithms used at different steps of the documents processing. Section 3 describes the implementation of the system and preliminary results. In section 4, we discuss the ongoing work and present future development ideas. Finally, we present our conclusions in section 5.

## 2. ADAC system

The ADAC concept could be implemented according to different configurations. One of the configurations we privileged in this work is based on the concept of a desktop knowledge management and decision support tool. This configuration is represented by Figure 2.

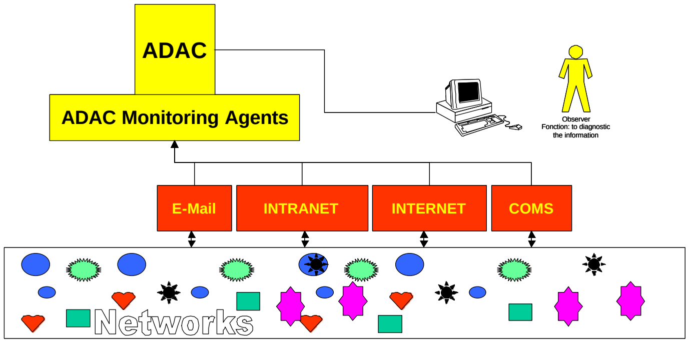


Figure 2: ADAC Retained Configuration

In this configuration, ADAC' agents continually monitor the flow of information in different media (e.g. e-mail, Intranet, Internet, Voice/Electronic COMS). Once a document has been intercepted by the recovery agent (Figure 4), it is copied into ADAC database and processed through the different steps as shown by Figure 3.

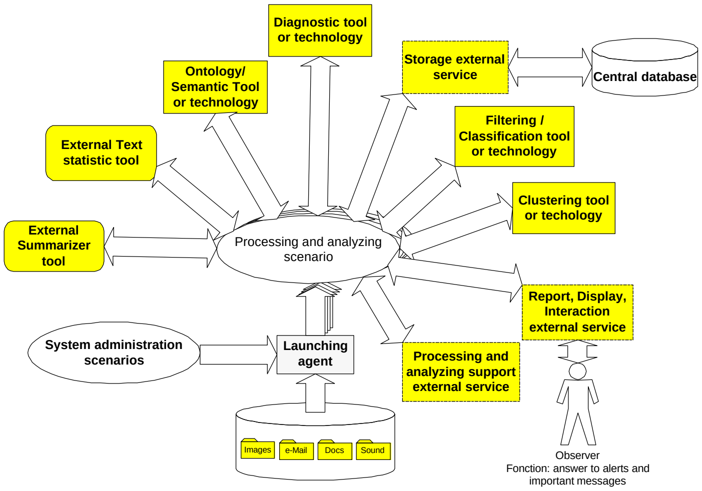

Diagnostic tool or technology
Storage external service
Central database
Ontology/ Semantic Tool or technology
Filtering / Classification tool or technology
External Text statistic tool
Clustering tool or techology
External Summarizer tool
Processing and analyzing scenario
Report, Display, Interaction external service
System administration scenarios
Launching agent
Processing and analyzing support external service
Images
e-Mail
Docs
Sound
Observer
Fonction: answer to alerts and important messages

Figure 3: ADAC Processing and Analyzing Scenario

The ADAC recovery agent could be configured as represented by Figure 4. In this configuration, the agent is able to deal with document having been produced into different formats (e.g. audio, image, text). The document is then processed according to the approaches/algorithms described below.

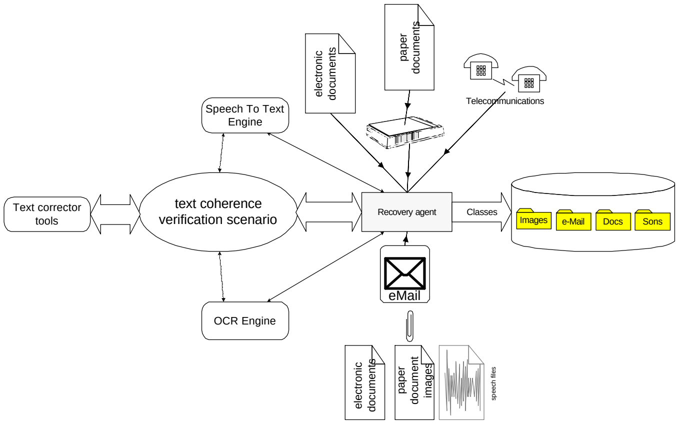
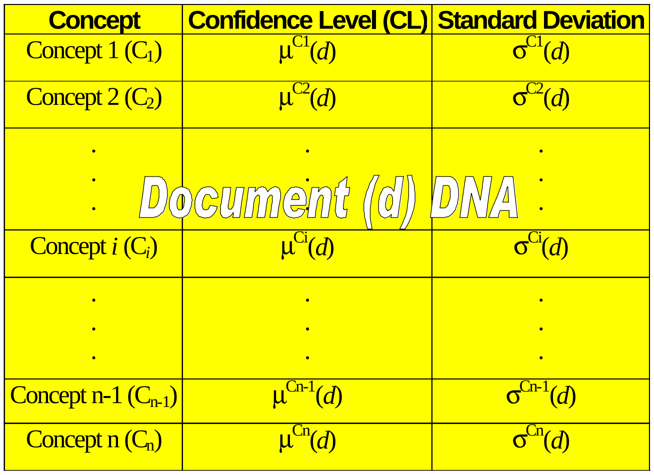


Figure 4: ADAC Recovery Agent

## 2.1 *Document processing steps used in ADAC*

Extracting relevant descriptors from free-text electronic documents is a problem that requires the use of natural language processing techniques. Statistic analysis of documents consists of extracting a set of concepts or attributes that characterize the text content based on statistical parameters (e.g. number of occurrences of words). Different statistical methods have been proposed in the domain (e.g. Latent Semantic Indexing). However, purely statistical methods may lead to text descriptors that do not really reflect the semantics of processed documents.

Any document intercepted by ADAC agents is summarized and its most pertinent concepts are extracted. Then, using the ontology and its related semantic networks, we extract the document’s DNA that consists of statistical measurement on the document. A document’s DNA (text DNA) could be represented as shown in Figure 5. For each concept we measure a confidence level indicating that the concept is represented within the document with a standard error. This matrix will then be attached to the document through its journey within ADAC. It is also possible to export this data to other applications.


Document (d) DNA

| Concept | Confidence Level (CL) | Standard Deviation |
|---|---|---|
| Concept 1 ($C_1$) | $\mu^{C1}(d)$ | $\sigma^{C1}(d)$ |
| Concept 2 ($C_2$) | $\mu^{C2}(d)$ | $\sigma^{C2}(d)$ |
| .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| Concept *i* ($C_i$) | $\mu^{Ci}(d)$ | $\sigma^{Ci}(d)$ |
| .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| Concept n-1 ($C_{n-1}$) | $\mu^{Cn-1}(d)$ | $\sigma^{Cn-1}(d)$ |
| Concept n ($C_n$) | $\mu^{Cn}(d)$ | $\sigma^{Cn}(d)$ |

Figure 5: Document DNA

This DNA is then used to diagnose pattern or to retrieve specific concepts related documents. It is also used to classify and cluster documents according to their contents.

### 2.1.1 *Summarizing*

The summarizer tool retained for ADAC is Copernic – Summarizer Server. The main features used in ADAC are the extraction of the key concepts and the summarizing of documents. The key concepts and the summary are saved as document properties and can be shown to the operator.

### 2.1.2 *Semantic analysis*

An ontology [Gruber, 1993] is a formal specification of a shared understanding of a domain of interest, described by concepts and relations between concepts. Providing a formal model of a domain through ontologies, or classification of terms through taxonomies has often been identified of potential utility to support information extraction from texts or for automated document indexing. For example, WordNet, a large electronic lexical database publicly available [Felbaum, 1998] may be used to support information extraction or query formulation. But it is considered not suitable for processing texts in specific domains.

Recently, some preliminary experiments have been conducted to combine statistics and semantics for information extraction [Termier *et al*., 2001, Faure and Poibeau, 2000] using different methods. Our approach favors an ontology-based semantic analysis that consists of analyzing candidate concepts resulting from the statistical analysis from a semantic perspective by exploiting a domain ontology in order to restrict the document descriptors to the attributes that semantically characterize the text (e.g. remove poorly meaningful words).

To demonstrate our approach and techniques, we have restricted our experiment to a particular specific domain, the terrorism. In this context, we have chosen a document corpus from various open sources and build a baseline ontology about terrorism that organizes concepts of the domain in a hierarchy of concepts. Concepts at the first level in the ontology consist of the main categories of concepts for this domain. Level 2 refines the concepts of level 1 by providing more specific concepts in the hierarchy, most of them being linked by a "is-a" relationship. At each level, specific semantic expressions are attached to concepts, similar to subcategorization frames (e.g. in the ASIUM system [Faure and Poibeau, 2000]) to guide the semantic processing. A semantic search engine is used to search for semantic similar expressions in documents being processed. This allows the system to refine the semantic analysis of the document and thus provides a fine-grained documents categorization. For example, from the concept *bomb* extracted from a document, we could deduce through this process whether the document is about "*bomb construction*" or "*bomb explosion*". Figure 6 illustrates the baseline terrorism ontology we have built from resources found on the Web for the purpose of our experiment (e.g. [NATO, 2002]).

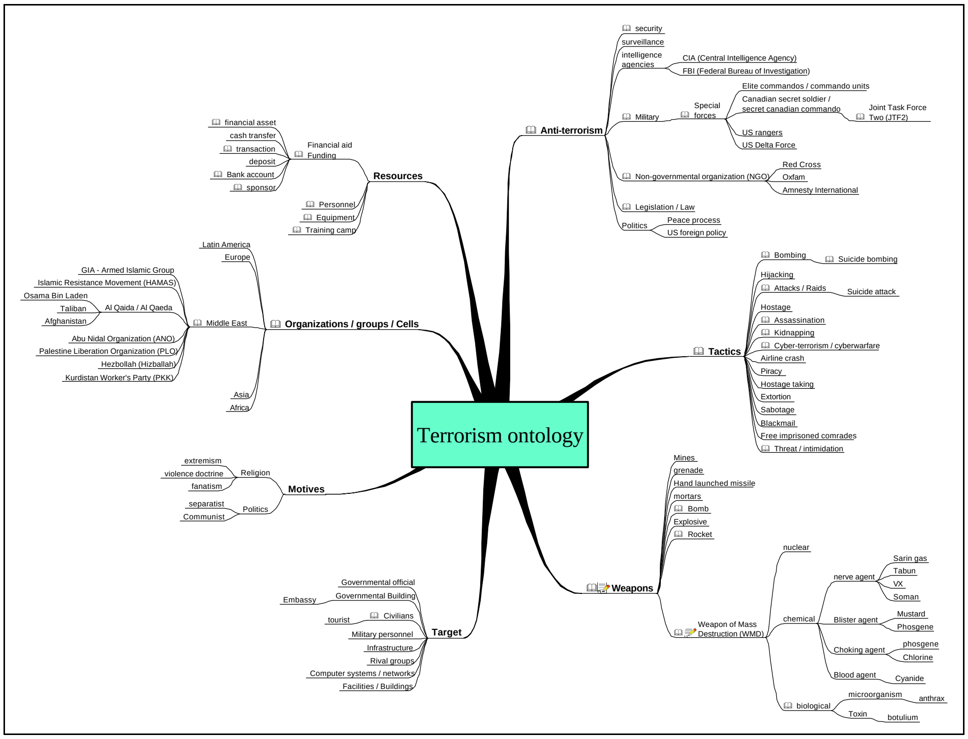


Figure 6: Hierarchy of concepts in the terrorism ontology

### 2.1.3 *Diagnostic*

Every document is analyzed through the diagnostic tool to determine its relevance relatively to a specific query (or a set of queries). The diagnostic is based on the idea of continuous query. Documents and queries are represented using the DNA matrix-space model. The user, using the *Diagnostic* interface, specifies the keywords or concepts to be retrieved in documents. The diagnostic is based on the similarity between the query (*q*) and the processed document (*d*). This similarity is computed by comparing the two DNAs. A fuzzy similarity index is then computed as follows [Belacel, 2000]:

$$\tilde{I}(d,q)=\sum_{j=1}^{m}\tilde{I}_j(d,q)=\sum_{j=1}^{m}(w_j^h \times C_j(d,q)\times(1-D_j(d,q)))$$

The *concordance* and the *discordance indexes* ($\mathrm{C_j(d,q)\,and\,D_j(d,q)}$) are computed at the concept level, by comparing the scores, between the request $q$ (formulated by the user) and the processed document *d*. These comparisons, as for the classification (section 2.1.4), are made on the averages of the confidence degrees for the *concordance*, and on deviations errors for the *discordance*. $\mathrm{C_j(d,q)}$ and $\mathrm{D_j(d,q)}$ take values between 0 and 1. $w_j$ are the weight-coefficients associated with the query concepts or keywords, reflecting the relative importance of each concept retrieved in the document; 0 $w_j$ 1 and $\sum_j w_j$ =1.

It is then easy to use α-cut concept to validate the diagnostic. For example, it is possible to consider a fixed α-cut threshold (e.g., 60%) and each time the fuzzy similarity index is greater than this threshold, an appropriate action (specified by the user) is triggered.

### 2.1.4 *Classification/Filtering*

The classification problem can be seen as the assignment of each document to a pre-defined category or class. The classes could be characterized by prototypes. The classes could be linked hierarchically or defined without any order relationship. The proposed algorithm considers any of these two options. The user can fix some prototypes for each class using some well-known examples where he is satisfied with their classification. The system learns from those past examples and automatically builds on the profiles of the prototypes. An appropriate DNA represents each prototype's profile. Any incoming document's DNA is then compared with each prototype's DNA of each class as shown by Figure 7.

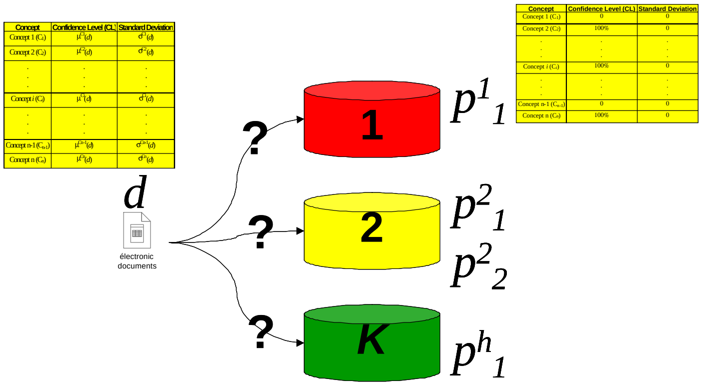

Concept | Confidence Level (CL) | Standard Deviation
Concept 1 (C1) | μC1(d) | σC1(d)
Concept 2 (C2) | μC2(d) | σC2(d)
Concept i (Ci) | μCi(d) | σCi(d)
Concept n-1 (Cn-1) | μCn-1(d) | σCn-1(d)
Concept n (Cn) | μCn(d) | σCn(d)
d
électronic documents
?
?
?
1
2
K
p1 1
p2 1
p2 2
ph 1
Concept | Confidence Level (CL) | Standard Deviation
Concept 1 (C1) | 0 | 0
Concept 2 (C2) | 100% | 0
Concept i (Ci) | 100% | 0
Concept n-1 (Cn-1) | 0 | 0
Concept n (Cn) | 100% | 0

Figure 7: Classification Process

The classification/filtering process is inspired from the PROAFTN[2] method [Belacel, 2000]. According to that special assignment process, a fuzzy similarity index is then computed as follows:

At a local level, (i.e. at the concept $j$ level)

$$\tilde{I}_j(d, p_i^h) = w_j^h \times C_j(d, p_i^h) \times (1 - D_j(d, p_i^h))$$

At a global level, (i.e. for the whole document)

$$\tilde{I}(d, p_i^h) = \sum_{j=1}^{m} \tilde{I}_j(d, p_i^h) = \sum_{j=1}^{m} (w_j^h \times C_j(d, p_i^h) \times (1 - D_j(d, p_i^h)))$$

with $m$, the total number of concepts retrieved in the document and $w_j^h$ the weight affected to the concept $j$ according to its position in the hierarchy (Figure 8).

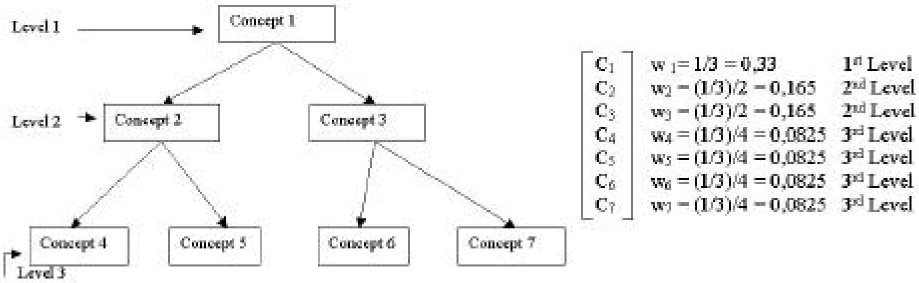

Level 1
Level 2
Level 3
Concept 1
Concept 2
Concept 3
Concept 4
Concept 5
Concept 6
Concept 7
C1 w 1= 1/3 = 0,33 1st Level
C2 w2 = (1/3)/2 = 0,165 2nd Level
C3 w3 = (1/3)/2 = 0,165 2nd Level
C4 w4 = (1/3)/4 = 0,0825 3rd Level
C5 w5 = (1/3)/4 = 0,0825 3rd Level
C6 w6 = (1/3)/4 = 0,0825 3rd Level
C7 w7 = (1/3)/4 = 0,0825 3rd Level

Figure 8: Concepts Importance Weighting

[2] PROAFTN : **PRO**cédure d'**A**ffectation **F**loue pour le **T**ri **N**ominal

As mentioned in the diagnostic process, the *concordance* and the *discordance indexes* ($C_j(d,p_i^h)$ and $D_j(d,p_i^h)$) are computed at the concept level, by comparing the scores, between the prototype $i$ (of a certain class $h$) and the processed document *d*. Comparisons are made on the averages of the confidence degrees for the *concordance*, and on deviations errors for the *discordance*. For more details, consult [Tiliki, 2002].

As $\mathrm{C_j(d,q)}$ and $\mathrm{D_j(d,q)}$ take values between 0 and 1, the local indifference index $\mathrm{\tilde{I}_j(d,p_i^h)}$ also takes values between 0 and 1. Thus the global indifference index will vary as: $0 \leq \tilde{I}(d,p_i^h) \leq m$.

In order to express $\tilde{I}(d,p_i^h)$ as a percentage handle scores, we divide it by the total number $m$ of concepts retrieved in the document. The score characterizing the membership of a document to a given class is then defined as the *membership degree*, given by:

$$I(d,p_i^h) = \left(\frac{\tilde{I}(d,p_i^h)}{m}\right) \times 100\ (\%)$$

If many prototypes describe a class, the *membership function* of a document to a specific class is obtained by considering the maximum similarity to any of these prototypes. This is given by:

$$\mathrm{\tilde{m}(d,C^h) = max\{I(d,p_1^h), I(d,p_2^h),..., I(d,p_i^h),....I(d,p_{L_h}^h)\}}$$

and the decision to assign exclusively a document to a specific class is based on the following rule:

$$\mathrm{\tilde{m}(d,C^h) = max\{\tilde{m}(d,C^1), \tilde{m}(d,C^2),...., \tilde{m}(d,C^l),.....\tilde{m}(d,C^k)\} \Leftrightarrow d \in C^h}$$

At the end of the classification/filtering process, a *forced classification* is carried out to link the document to the upper classes (or concepts) in the hierarchy. This facilitates correlation of information between the document's content and other logically related concepts.

### 2.1.5 *Clustering*

Clustering is an unsupervised learning method of classification that seeks to identify similar objects in a multi-dimensional space [Everitt, 1993]. The difference between the clustering and the classification can be seen in the pre-definition of the categories or the classes. The clustering problem consists of finding the best partition of a set of documents into sub-sets (categories) that best cluster the documents. The problem can be decomposed into a partitioning problem (creating categories) and then an assignment problem of the document to those categories. By definition, the clustering algorithm requires a database containing many documents. This algorithm could be executed each time the number of documents intercepted reaches a certain peak.

Let each $C^l$ represents the cluster $l$ with cardinality $n_l$, i.e., $|C^l|= n_l$. We can define the similarity within a cluster as the minimum similarity between any two elements of this cluster (intra-similarity). This can be formulated as Sim($C^l$) = $\check{S}$ ($C^l$) = min $I(d_i,d_j)$; $\forall$ $d_i$ , $d_j \in C^l$.

Let $\alpha_0$ be a similarity threshold fixed by the user if desired. Let also $\Delta_{lh}$ be the distance between two distinct clusters $C^l$ and $C^h$ (or inter-similarity) obtained using the Jaccard coefficient or the cosine function [Salton and Mc Gill, 1983]:

$$\Delta_{\mathrm{lh}} = \mathrm{J}(\mathrm{C}^{\mathrm{l}},\mathrm{C}^{\mathrm{h}}) = \frac{\left|\mathrm{C}^{\mathrm{l}} \cap \mathrm{C}^{\mathrm{h}}\right|}{\left|\mathrm{C}^{\mathrm{l}}\right| + \left|\mathrm{C}^{\mathrm{h}}\right| - \left|\mathrm{C}^{\mathrm{l}} \cap \mathrm{C}^{\mathrm{h}}\right|} \qquad \text{or } \Delta_{\mathrm{lh}} = \frac{\left|\mathrm{C}^{\mathrm{l}} \cap \mathrm{C}^{\mathrm{h}}\right|}{\sqrt{\left|\mathrm{C}^{\mathrm{h}}\right| \times \left|\mathrm{C}^{\mathrm{l}}\right|}}$$

The originality of the clustering algorithm proposed in this work lies in the multicriteria aspect where the objectives are the number of clusters, *a priori* unknown, to be minimized, the intra-cluster similarities and the inter-clusters distance to be maximized. The multi-objective constrained problem is stated as follows:

$$\begin{cases} \max \quad \breve{S}(\mathrm{C}^{\mathrm{j}}) \\ \min \quad \mathrm{k} \\ \max\limits_{\mathrm{j,h}} \quad \Delta_{\mathrm{jh}} \\ \text{s.t.} \\ \breve{S}(\mathrm{C}^{\mathrm{j}}) \geq \alpha_0;\ \mathrm{j} = 1,\ldots,\mathrm{k} \\ \left(\mathrm{n}_{\mathrm{j}} = \left|\mathrm{C}^{\mathrm{j}}\right|\right) \geq \mathrm{n}_0;\ \mathrm{j} = 1,\ldots,\mathrm{k} \\ \mathrm{k} \leq \mathrm{k}_0 \\ \Delta_{\mathrm{jh}} \geq \Delta_0;\ \mathrm{j} = 1,\ldots,\mathrm{k} \end{cases}$$

Where $k_0$ is a prefixed maximum number of clusters that could be fixed by the user or an external agent. The clustering algorithm consists of three stages: initialization, clustering improvement using genetic algorithms to maximize $\check{S}$ ($C^l$), merging clusters to minimize $k$ and maximize $\Delta_{lh}$.

*Stage 1*: Initial clustering

This is done by considering each document $d_i$ as an initial cluster and the closest documents $d_j$ are assigned to this cluster if their fuzzy similarity $I(d_i , d_j)$ are greater than a threshold $\alpha_0$. Let us denote this initial number of clusters $k$(t=0) = N.

*Stage 2*: Clustering improvement using genetic algorithms (GA)

GA is a meta-heuristic working with a population of solutions encoded into chromosomes and characterized by fitness, which returns a quantitative measure of their “goodness” [Goldberg, 1989]. In this work, a cluster is a chromosome, which encodes the membership of each document. It is a binary N-integer string where the $i^{th}$ document is set to one if it is present or to zero if not. Fitness of each cluster, defined by $\check{S}\ (C^l)$, must be maximized. A linear static scaling method of this objective function is used to ensure that the population did not become dominated by the descendants of a single super-fit chromosome, and later, if the population average fitness may be close to the fitness of the best chromosome, competition among chromosomes is not stochastic but based on the principle of the survival of the fittest [De Jong, 1976].

Evolution of the population, i.e. improvement of the fitness of solutions, is done using two principal operators. The first operator is the crossover that combines parts of the fittest chromosomes of the previous generation to generate new individuals. Two types of crossover are used alternatively: the uniform crossover [Syswerda, 1989] and the partially mapped crossover [Goldberg and Lingle, 1985]. The second operator is the mutation that introduces new information at random with a low probability by changing the value of one single bit from one to zero and *vice versa*, at a randomly chosen position in the string.

The selection of clusters for reproduction, crossover and mutation is done using the stochastic remainder selection without replacement [De Jong 1976; Goldberg, 1989]. The probability of selection is calculated by $p_s(C^l) = \tilde{S}(C^1) / \sum_{l=1}^{k(t)} \tilde{S}(C^1)$.

*Stage 3*: Merging clusters

At this stage of the procedure, we try to reduce the number of clusters obtained from the second stage while maximizing the distance between clusters. This is done by evaluating inter-similarity $\Delta_{lh}$ between all the clusters C and merging the closest ones. The new number of clusters is then $k(t+1)$. The procedure is repeated from stage 2.

The clustering is stopped if the number of iterations reached the maximum or if the number of clusters cannot be reduced without altering the intra-cluster similarity.

## 3. Implementation and results

ADAC has been implemented by the Quebec based Intell@xiom inc. It has been developed within the IntellStudio environment and integrates several external services and algorithms as shown by Figure 9. IntellStudio allows the interoperability of heterogeneous systems, COTS, GOTS and supports all major known communication protocols. It also offers tools to easily integrate knowledge and analytical based decision rules into the document processing flow.

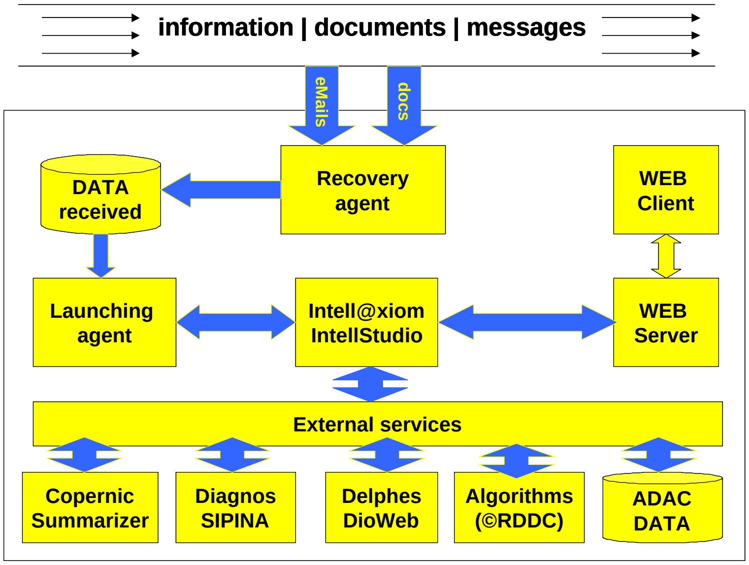


Figure 9: ADAC Implementation Concept

The Copernic's Summarizer is used to generate a comprehensive summary along with the ten most representative concepts of the document. The SIPINA Diagnos tool is used as a statistical tool. It offers several data mining tools and knowledge extracting tools (structured or unstructured). The DioWeb Delphes is a semantic search engine. It is used within ADAC as a semantic search engine (based on ontology) in documents. The Algorithms external service invokes the three algorithms: diagnostic, classification, and clustering. The architecture showed in Figure 10 is a detailed view of the ADAC Implementation Concept shown in Figure 9.

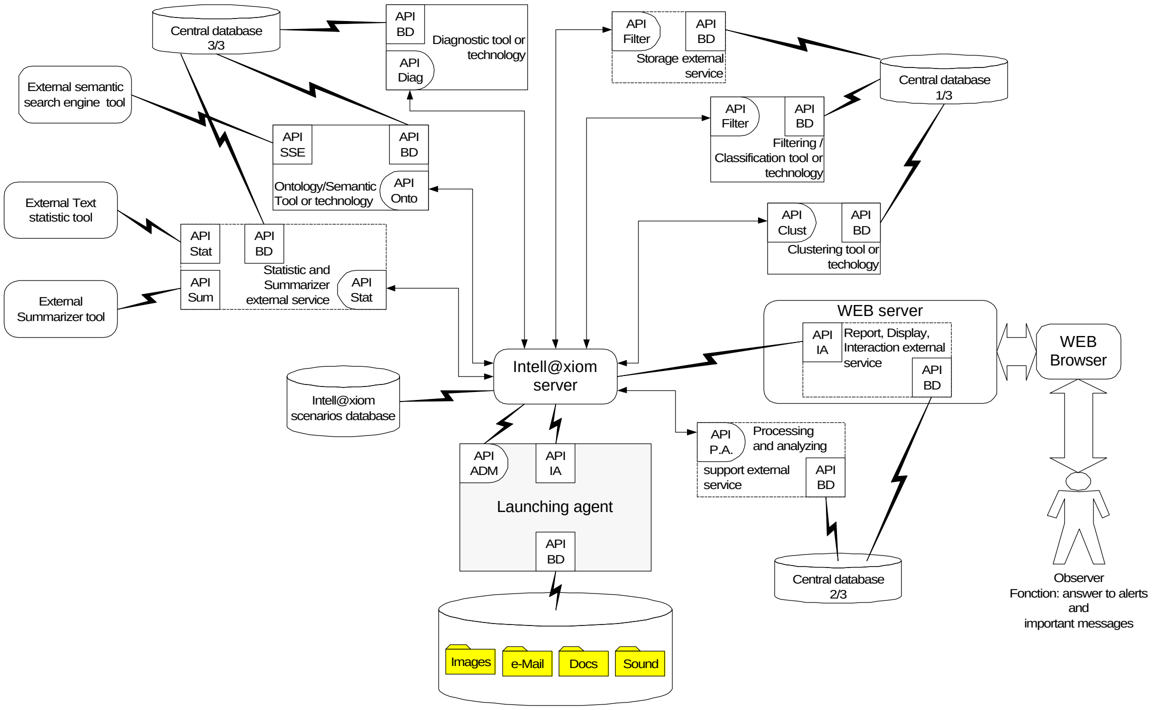

Central database 3/3
API BD
Diagnostic tool or technology
API Diag
API Filter
API BD
Storage external service
Central database 1/3
External semantic search engine tool
API SSE
API BD
Ontology/Semantic Tool or technology
API Onto
API Filter
API BD
Filtering / Classification tool or technology
External Text statistic tool
API Stat
API BD
Statistic and Summarizer external service
API Sum
API Stat
API Clust
API BD
Clustering tool or techology
External Summarizer tool
WEB server
API IA
Report, Display, Interaction external service
API BD
WEB Browser
Intell@xiom server
Intell@xiom scenarios database
API ADM
API IA
Launching agent
API BD
API P.A.
Processing and analyzing support external service
API BD
Central database 2/3
Observer
Fonction: answer to alerts and important messages
Images
e-Mail
Docs
Sound

Figure 10: ADAC Architecture

Figure 11 shows an example of user interfaces to access ADAC within the desktop knowledge management and decision support configuration. In this configuration, the user can access to multiple documents, as well as to the classification and clustering results. Advanced search engines are available to browse the database and create persistent queries for diagnostic. The alerts will be displayed and external actions could be pre-programmed. A map viewer allows associating documents with specific geographical areas in the world.

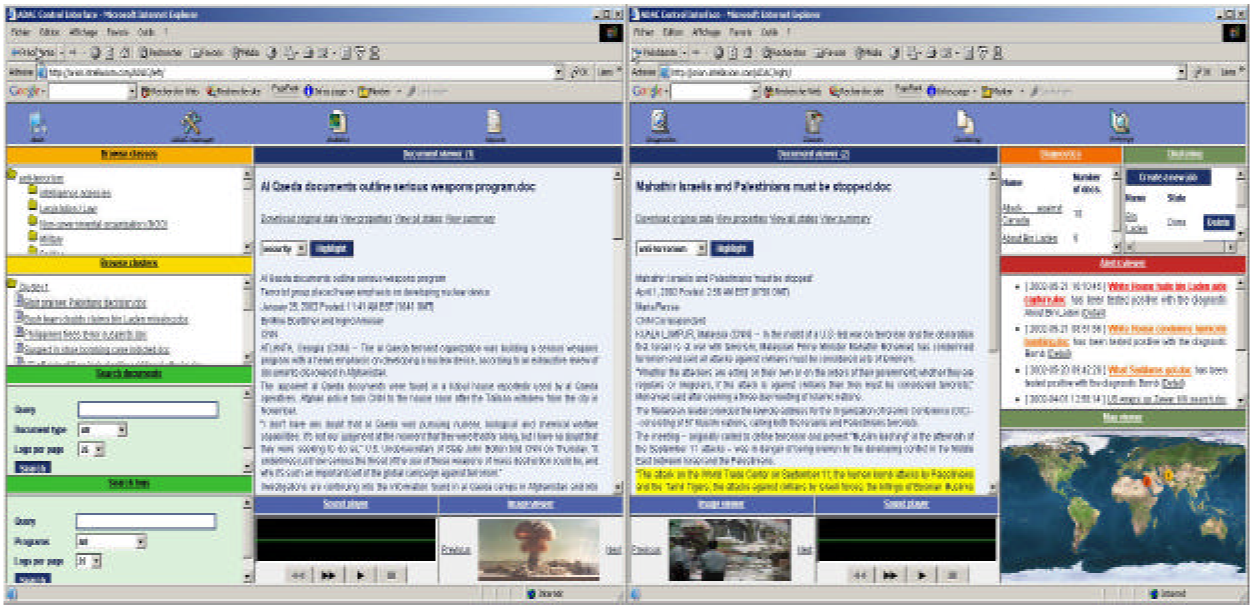


Figure 11: ADAC Interfaces (example)

## 4. Ongoing work

The ADAC prototype developed shows how putting together knowledge management and decision analysis concepts could lead to an advanced tool to support military organisations coping with information overload while minimising risks. ADAC proposes advanced concept demonstrators for unstructured and structured information management and offers several decision support tools to help understanding information content and gaining situation awareness.

The different algorithms developed require more fine-tuning and validation. In fact, we are going to perform several experimentations to evaluate the algorithms performance, faithfulness, effectiveness and efficiency. Other clustering and classification algorithms will be also developed and tested.

The ontology and the semantic networks are going to be reviewed with subject-matter experts at the NDCC and adapted to reflect their needs. More advanced semantic search engine will be used to generate document's DNA. The DNA itself will be improved and the measurement will be enhanced.

We are also looking to showcase other ADAC configurations to support other R&D projects. Figure 12 shows two different configurations. Figure 12.A represents a LAN/WAN filtering system that monitors all the traffic, structures the information and triggers alerts. Figure 12.B shows ADAC used as an input to a document management system (e.g. JIIMS).

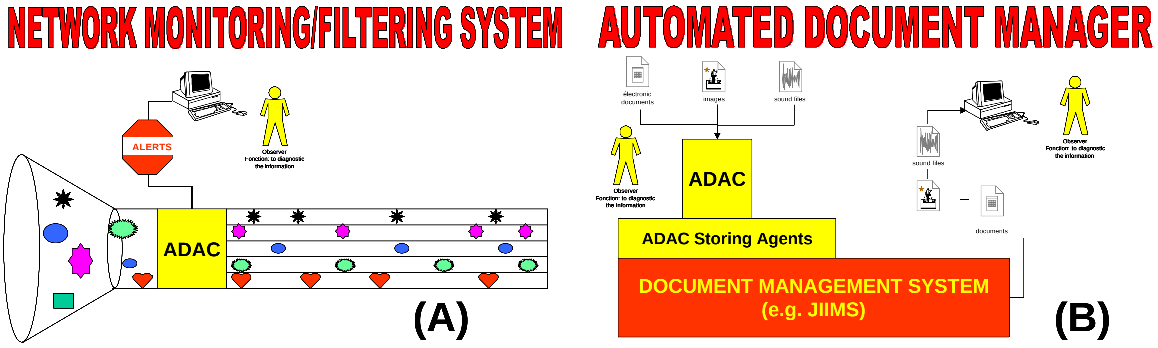

NETWORK MONITORING/FILTERING SYSTEM
ALERTS
Observer
Fonction: to diagnostic the information
ADAC
(A)
AUTOMATED DOCUMENT MANAGER
électronic documents
images
sound files
Observer
Fonction: to diagnostic the information
ADAC
sound files
documents
Observer
Fonction: to diagnostic the information
ADAC Storing Agents
DOCUMENT MANAGEMENT SYSTEM
(e.g. JIIMS)
(B)

Figure 12: Other ADAC's Configurations

## 5. Conclusion

Exploiting the relationships between the concepts in the ontology could enhance the semantic processing of documents. Building an ontology for a specific domain is a time-consuming task. Our approach should be enriched in order to be able to learn new concepts (ontology learning).

ADAC is a prototype that showed the feasibility of an advanced automated document manager, analyzer, classifier and diagnosis tool. The ADAC uses the concept of document's DNA that is a matrix extracted using advanced ontology/semantic networking tools. Ontology and semantic networks should be then carefully developed with the end-users and should be validated with subject matters experts.

We have proposed several powerful and original algorithms to perform diagnoses, classification and clustering. Those algorithms should be extensively tested and validated. The IntellStudio development environment is a powerful decision support systems development tool. It offers flexible and easy access to external services (integration of COTS & GOTS), user-friendly interfaces supported by graphical modelling, and easy integration of knowledge and analytical based rules.

It is important to integrate automated learning algorithms to improve ADAC flexibility and improve its usefulness to effectively support end-users. It is also important to integrate services that process synonyms, advanced semantic networks, image processing, speech to text, intelligent Character Recognition. We are also going to investigate ways to improve external actions like the visualization of the information, generating alerts in different format including sending wireless messages.

Humans always put together pieces of information from different sources to learn new concepts and understand documents' contents. It is recommended to pursue research into advanced ways to network documents and improve learning by putting together pieces of information from different sources. Bayesian networks are seen as a potential way to implement such concepts.

## 6. **References**

[Belacel, 2000] N. Belacel, Multicriteria assignment method PROAFTN: methodology and medical assignment, *Eur. J. Oper. Res.*, 125: 175-183, 2000.

[De Jong, 1976] A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD Thesis, University of Michigan, USA, 1976.

[Everitt, 1993] B.S. Everitt, *Cluster Analysis*, London: Edward Arnold, 1993.

[Faure and Poibeau, 2000] D. Faure, T. Poibeau, First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX, in: *the proceedings of the 14th European Conference on Artificial Intelligence*, ECAI'2000, Berlin, Germany, 2000.

[Felbaum, 1998] C. Felbaum, *WordNet: an electronic lexical database*, Cambridge, Massachussets and London, England, The MIT Press, 1998.

[Goldberg, 1989] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, M.A., 1989.

[Goldberg and Lingle, 1985] D.E. Goldberg, R. Lingle, Alleles, loci and the traveling salesman problem, in: *the Proceedings of the First International Conference on Genetic Algorithms*, J. Grefenstette (Ed.), Lawrence Erlbaum Associates, Hilsdale, NJ, 1985.

[Gruber, 1993] T. Gruber, A translation approach to portable ontology specifications, *Knowledge acquisition*, 5 :199-220, 1993.

[Salton and Mc Gill, 1983] G. Salton, M.J. McGill, *Introduction to Modern Information Retrieval*, NY: Mc Graw Hill, 1983.

[Syswerda, 1989] G. Syswerda, Uniform crossover in genetic algorithms, in *the Proceedings of the Third International Conference on Genetic Algorithms*, J. Schaffer (Ed.), Morgan Kaufmann publishers, Sam Mateo, CA, 1989.

[NATO, 2002] NATO/RTO, *RTO Combating terrorism Workshop Report*, April 2002, http://www.rta.nato.int/ctreport/CTWSReport.pdf.

[Termier et al., 2001] A. Termier, M.-C. Rousset, M. Sebag, Combining Statistics and Semantics for Word and Document Clustering, in: *the proceedings of IJCAI 2001*, Workshop on « Ontology Learning », 2001.

[Tiliki, 2002] K. Tiliki, *Development of an automatic method of documents classification,* ADAC project, Université Laval, Québec, Canada, 2002. (to be submitted)