Building A Framework For Situation Awareness

J. Salerno
AFRL/IFEA
AF Research Lab
Rome, NY 13441-4114
USA
salernoj@rl.af.mil

M. Hinman
AFRL/IFEA
AF Research Lab
Rome, NY 13441-4114
USA
hinmanm@rl.af.mil

D. Boulware
AFRL/IFEA
AF Research Lab
Rome, NY 13441-4114
USA
boulward@rl.af.mil

Abstract – *There has been much activity over the past two decades in developing conceptual models under the titles of data fusion and situation awareness. In this paper we will explore the two most popular models and show how they complement each other in developing an overall framework for situation awareness. We will also demonstrate how this framework has been applied to a sample "monitoring" problem.*

Keywords: Higher Level Fusion, Situation Awareness, Indications & Warning (I&W), SA Framework, Data Mining, Community Generation, Natural Language Extraction

1 Introduction

Over the years, more than thirty fusion models have been proposed and countless research initiatives and personnel have attempted to define these models in great detail. However, no model has become as influential in Data Fusion as the Joint Director's of Laboratories (JDL). As shown in Figure 1, and described in [9], the JDL model has five levels: Level 0 – Sub-Object Data Assessment; Level 1 – Object Assessment; Level 2 – Situation Assessment; Level 3 – Impact Assessment; and Level 4 – Process Refinement.

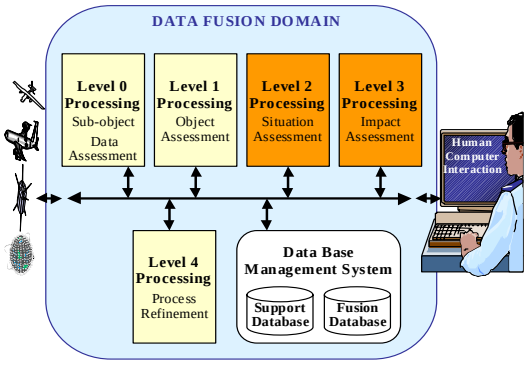

Fig. 1. JDL Fusion Model.

A stream of data enters the model at level 0, Sub-Object Data Assessment. Level 0 provides physical access to the raw bits or signal. In addition, estimation and prediction of the existence of an object is performed based on pixel or signal level data association and characterization.

Objects are correlated and tagged over time in an attempt to build tracks and to perform object identification during level 1 processing, or Object Assessment. During Situation Assessment, or level 2 processing, the knowledge of objects, their characteristics, relationships with each other and cross force relations are aggregated in an attempt to understand the current situation. Previously discovered or learned models generally drive this assessment. After Situation Assessment, the impact of the given situation must be assessed (Level 3 – Impact Assessment). The impact estimate can include likelihood estimates and cost/utility measures associated with the potential outcomes of a player's planned actions. The final level, Process Refinement, provides a feedback mechanism to each of the other layers, including the sensor itself. To date, research driven by the JDL model has concentrated on sensor level (0 and 1) object identification and tracking algorithms and in developing algorithms to perform model assessment.

While the JDL provides a functional model for the data fusion process, it does not model it from a human perspective. Endsley [2] provides an alternative to the JDL model that addresses Situation Awareness from this viewpoint (i.e., Mental Model). Her model has two main parts: the core Situation Awareness portion and the various factors affecting Situation Awareness. The core portion follows Endsley's [3] proposition that Situation Awareness has three levels of mental representation: perception, comprehension, and projection. The second and much more elaborate part describes in detail the various factors affecting Situation Awareness. Endsley defines Situation Awareness as a state of knowledge that results from a process. This process, which may vary widely among individuals and contexts, is referred to as Situation Assessment, or as the process of achieving, acquiring, or maintaining Situation Awareness. The three levels of Situation Awareness as proposed by Endsley are summarized in Figure 2.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Building A Framework For Situation Awareness	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) **Air Force Research Laboratory,AFRL/IFEA,Rome,NY,13441-4114**	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
7th International Conference on Information Fusion, 28 Jun ? 1 Jul 2004, Stockholm, Sweden.

14. ABSTRACT
There has been much activity over the past two decades in developing conceptual models under the titles of data fusion and situation awareness. In this paper we will explore the two most popular models and show how they complement each other in developing an overall framework for situation awareness. We will also demonstrate how this framework has been applied to a sample "monitoring" problem.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

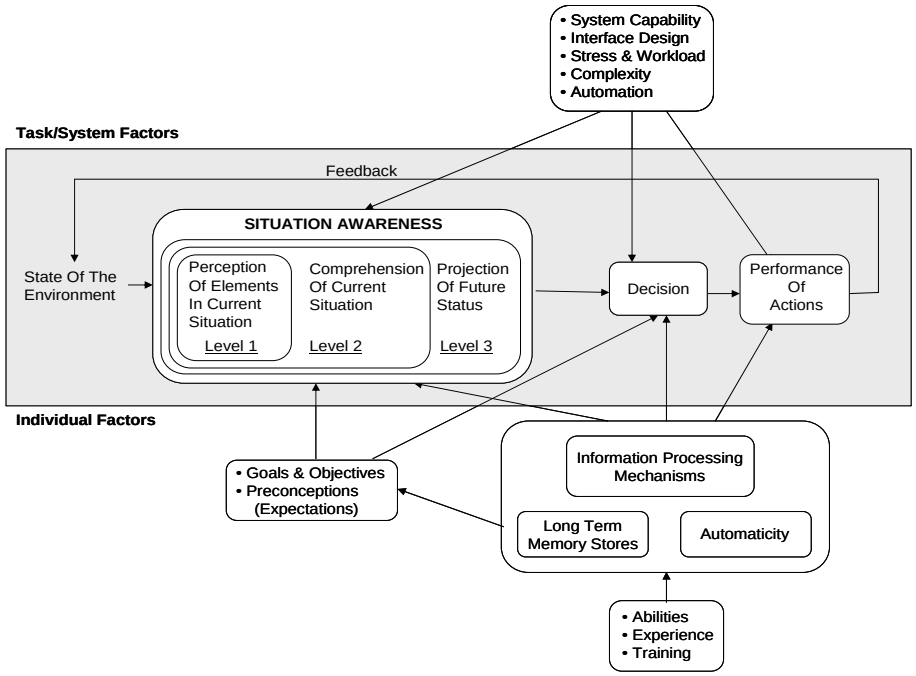

Fig. 2. Endsley's Situation Awareness Model.

According to Endsley, Situation Awareness begins with Perception. Perception provides information about the status, attributes and dynamics of the relevant elements in the environment. It also includes the classification of information into understood representations and provides the basic building blocks for comprehension and projection. Without a basic perception of important information, the odds of forming an incorrect picture of the situation increase dramatically.

Comprehension of the situation encompasses how people combine, interpret, store, and retain information. Thus, it includes more than perceiving or attending to information; it includes the integration of multiple pieces of information and a determination of their relevance to the underlying goals. Comprehension yields an organized picture of the current situation by determining the significance of objects and events. Furthermore, as a dynamic process, comprehension must combine new information with already existing knowledge to produce a composite picture of the situation as it evolves. Endsley notes that the ability to forecast future events marks decision-makers that have the highest level of Situation Awareness and refers to this as Projection. Situation Awareness refers to the knowledge of the status and dynamics of the situational elements and the ability to make predictions based on that knowledge. McGuinness and Foy [8] extended Endsley's Model by adding a fourth level, which they called Resolution. This level provides awareness of the best path to follow to achieve the desired outcome to the situation. Resolution results from drawing a single course of action from a subset of available actions. McGuinness and Foy believe that for any fusion system to be successful, it must be resilient and dynamic. It must also address the entire process; from data acquisition to awareness, prediction and the ability to request elaboration or additional data. McGuiness and Foy put Endsley's model and their model into perspective with an excellent analogy. They state that Perception is the attempt to answer the question "What are the current facts?"; Comprehension asks "What is actually going on?"; Projection asks "What is most likely to happen if…?" and Resolution asks "What exactly shall I do?" Another point to be made is that any proposed model should not promote a serial process, but rather a parallel one. Neither the JDL Model nor Endsley suggest otherwise. Each function (for example in Endsley's model: Perception, Comprehension, Projection and Resolution) happens in parallel with continuous updates provided to and from each other.

In the following sections we describe a framework that was developed based on the analysis of the two models. We present this framework as a process flow. After presenting the framework we show how it was used to build a functional demonstration.

2 Building A Framework

The process commences with the analyst defining the problem of interest. In many areas (e.g., Indications & Warning) much experience and knowledge has been obtained through history and various models have been developed which document this previous experience. The analyst begins with the adaptation of the model based on the specific concerns and parties involved (in terms of possible scenarios). This model defines what

pattern(s) we are interested in and indirectly what data/information the analyst requires to collect to develop an understanding of what is going on.

The Data Collection component receives the data requirements based on the model of interest and has the intelligence to determine what and where to gather the data and when to request updates. It then gathers this data, wraps it in a common document structure and publishes it along with metadata capturing various details such as when the information was collected, what source the information came from and the format of the data. Based on the format of the data, it may be necessary to parse it (e.g., formatted messages) or to extract relevant entities, relationships and events through the use of Natural Language Extractors. In any event, once events and relationships are obtained, there needs to be a cleansing process performed. The cleansing process removes redundant, incomplete and "dirty" data. It also deals with data transformations and aliases. The goal of this process is to provide an evidence database that is free from errors and contains perishability and confidence estimates. This evidence database forms what we defined as Endsley's "Perception". It should also be noted here that the collector is continuously gathering new data based on the problem at hand.

Perception also provides us with an interface to the sensor world. For this part we rely on the JDL model (levels 0 and 1) to provide us with an interface between real-time sensor data and observable objects/events. Because of the many limitations of computers to "understand" multi-media data, we must rely on many of the existing manual, human processes of exploitation. It is here we rely on the disciplines of Information Exploitation (IE). Simply put IE can be considered as a process to transform raw signals/data into formatted textual reports. An example here might provide better insight to the applicability and value of IE. Systems that automatically process imagery are rare and provide minimal capabilities. Let us consider Imagery Exploitation. Imagery is collected, Imagery Analysts (IA) or Photo Interpreters (PI) exploit imagery based on previous reports and imagery and the current image. One output of this process is a textual report or message describing any significant events in the image. These reports are then disseminated throughout the community through message handling systems. Most of these reports are structured for computer use. Based on this analogy and the state of the foreseeable future we focus our attention on textual input.

As the database is updated, Model Analysis tools are used to determine if any parts of the target models appear within the evidence. One way in accomplishing this is to build a graph from the database (which we refer to as the input graph) and compare the model (referred to as the target graph) using simple graph theory. Based on the analysis, any portions of the input graph that match the target graph are identified and provided to the analysts as alerts. This portion of the process defines the "Comprehension" portion of the model. That is, past knowledge (as defined by the analyst in terms of the model or target graph) when combined with the evidence (or perception) provides comprehension or understanding of the situation. Figure 3 provides an overview of the described process.

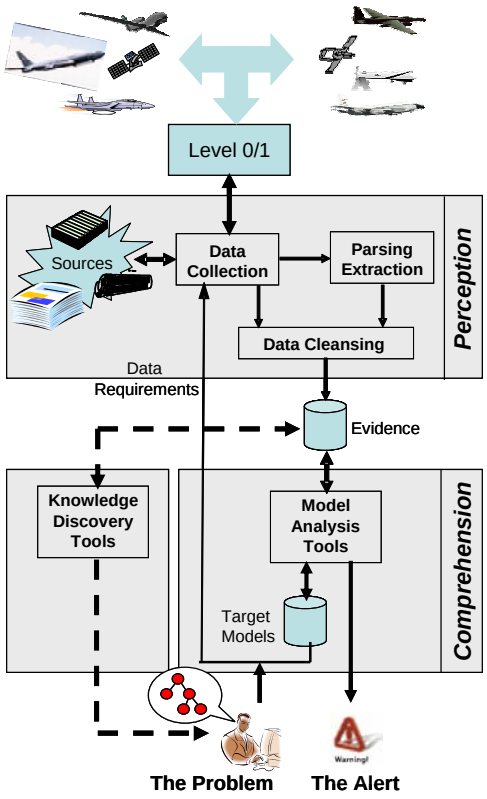

Fig. 3. Situation Awareness Framework.

In order to comprehend the current situation and its relevancy one must have some knowledge of similar situations that occurred in the past and relevant events currently occurring. If this prior knowledge does not exist, we need to learn or discover it. This knowledge can be captured as models which can be learned by deriving them through data sets and would include such concepts as activities, capabilities and group memberships. This area is what we have called Knowledge Discovery Tools. One of the major areas that fall under this topic is Data Mining.

2.1 Knowledge Discovery Tools

Predictive analysis requires information about past events and their outcomes. Much of the work in this area requires a predefined model built by subject matter experts, or substantial amounts of data to train model generation software to recognize patterns of activity. To date these models are manually intensive to construct, validate, and interpret. Algorithms are needed to provide efficient inferencing, reasoning, and machine learning procedures. Learning applications range from data mining programs that can discover general rules from large data sets to "knowledge assisted" hybrid

approaches aimed at accomplishing deeper levels of reasoning and pattern identification.

Witten, Frank & Gray [12] defined data mining as the extraction of implicit, previously unknown, and potentially useful information from data. The idea is to build computer programs that sift through databases automatically, seeking regularities or patterns. They go on to state that strong patterns, if found, will likely generalize to make accurate predictions on future data. Data mining techniques can be divided into two activities: (1) identifying patterns based on event associations which we refer to as pattern learning and (2) identifying groups based on similar activities which we refer to as community generation.

It is crucial that we thoroughly sift through archived data to look for the associations between entities at multiple levels of resolution. Pattern learning technologies serve to address this task by providing techniques that mine *relational* data. Pattern learning can be roughly described as the process of examining the relationships between entities in a database; the end-products of which are predictive models (statistical extrapolations) capable of describing what has been examined in terms of an abstract mathematical formalism (usually, a graph-theoretic construct). Relational data presents several interesting challenges:

- Relational learning must consider the neighborhood of a particular entity, and not just a singular record.
- Most learning is predicated on (usually false) assumptions of independent samples. Relational data does not meet this criterion.
- Data must be semi-structured to make learning possible. A query language must be developed to support the retrieval of data.

Jensen [5] states that the biggest concern in developing a pattern learner for situation awareness is the relatively low number of so-called "positive instances", turning the pattern learning process into an anomaly detection process. Problems such as these are often considered "ill-posed" in the computational learning community, and more often than not, partially invalid assumptions about the data must be made to correct for these conditions. If improperly handled, low rates of positive instances will completely confound the learning process, resulting in low-fidelity models, which produce high numbers of false positives/negatives. While the challenges are significant, so too is the potential payoff. Relational learning allows systems to exploit multiple tables in a database without the loss of information that occurs in a join or an aggregation [1]. The resulting discoveries may include predictive patterns that more accurately describe the world by utilizing entities' attributes as well as the relationships between entities in the learning process.

Missing and corrupted data are also prime sources of error. Numerical data is naturally a bit easier to work with, given the fact that we can interpolate. The lack of numerical descriptors for the type of archived data with which we often deal exacerbates the issue of missing items. Luckily, there has been a recent surge of research activity in the domain of relational learning addressing all of these issues.

Community generation and the class of problems it is trying to solve can be categorized as one of discerning group membership and structure. Under this topic two types of paradigms are being investigated: one where two parties and the activity type are given and one where only one party and one associated event is given. Zhang [14] describes the first class as bi-party and the later as uni-party.

Community generation algorithms will typically take events and relationships between individuals (whether implicit or explicit) and develop some correlation between them. This correlation value defines the strength of the link. Why are these models important to us? The models derived provide us insights into organizational structure and people of interest. Let us consider the first instance – organizational structure. Suppose that we have identified two groups whose structures are shown in Figure 4.

We can easily see from the models shown in Figure 4, that there is a key node in the model, which if removed or identified could have major impacts on the community. In this case, it could be a key individual within an organization. A second use of this information is the development of a behavioral model for the group. Knowing the individuals in charge of the group and "understanding" their behaviors or could facilitate more advanced modeling and simulation capabilities as well as direct surveillance efforts.

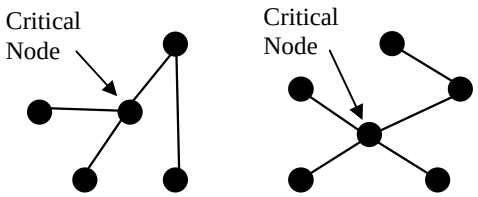

Fig. 4. Community Generated Models.

3 A Functional Demonstration

Thus far we have discussed many pieces of a large puzzle. To bring things back into perspective, we present a simple flow of the concept as shown in Figure 3. In the concept presented, there are two major flows –a background process and a "real" time process. It should be obvious by now that the concept that we have presented in this paper is model driven. The demonstration only integrates a subset of the components as described above. It also demonstrates only a top-down approach. We note here that we believe, depending on prior knowledge and past experience of the situation both the bottom-up (as presented by the JDL Model) and the top-down approach (as described by Endsley) are necessary. The demonstration begins with a well-known monitoring problem and was limited to the integration of user

generated models, data collection, document parsing/extraction and model analysis. We would like to reiterate again that the objectives of this first demonstration were twofold: (1) define the flow of information and an initial set of components to integrate, and (2) determine if the proposed architecture could support the concepts as described above. The last objective was of the greatest risk since none of the capabilities chosen were ever integrated with each other.

3.1 The Scenario

The scenario developed was based on the first Gulf War. One hundred and forty key events were identified from February 24, 1990, when Saddam Hussein threatened the Premier of Kuwait, through January 17, 1991 when the US began bombing Baghdad. The concern raised was Iraq's aggression towards its neighboring countries: Iran, Turkey and Kuwait. To fully investigate this scenario and a number of key technologies various components were loosely integrated via a publish and subscribe communications infrastructure, referred to as the Joint Battlespace Infosphere (JBI) [13]. The publish and subscribe mechanism was also utilized to develop a monitoring process. As each component receives work, they publish a management packet which is subscribed to by the monitoring component. The Graphical User Interface (GUI) alters its display as each component receives work. As each component publishes their activity, the monitor will visually display this activity by changing the color of the respective process to green. Also, any specific data corresponding to the activity is displayed in the textual window. This feature visually captures the interaction amongst various products and provides an interface for future interjection.

In addition to the monitoring capability, it was necessary to develop a mechanism that would allow complete control over what information is available as the scenario progresses. While collection is part of the process, it was necessary to develop an initial corpus which could later facilitate an evaluation of the system as a whole. A set of documents was collected via the Internet and formed this initial corpus. Each document title included the date published and the document was stored in a directory indicating which source it originated from. However, the desire was to introduce these documents into the system in a manner that was consistent with the way in which they would appear in reality. To address this requirement a program was written that generated scenario scripts which specified at what time each document should be made available to the system. The scenario scripts can also be generated with a compression factor that allows a day of scenario time to be reduced to a specified number of seconds in actual time. This feature was used to run a month long scenario in a matter of minutes; however, it could also be used to ensure the scenario develops in real time.

The documents still must be indexed after they have been introduced to the system to enable key word searches. To accomplish this, a background thread constantly monitors each source directory for new documents. If a specified time threshold has been exceeded since the last indexing and new documents are available the thread launches an indexer, SWISH-E [11] which generates a new index file for the altered source. While still capable of indexing a file immediately when introduced, this feature also enables us to simulate a lag in various information sources. While these features help establish a valid test environment they do not actually contribute to the situation awareness.

With a clearer understanding of the test environment, we may now begin to investigate the SA process in greater detail. The process begins by first defining the problem in terms of a model. The model is a simple acyclic graph specified in XML. A simple graphical interface allows an analyst to build various models and to submit them for execution. This is referred to as the activation of the model. Figure 5 shows a portion of the defined model. At the highest level is the warning or the concern to monitor. The problem is then divided into a number of general indicators or concepts. These general indicators can be further divided to provide more focused concepts and entities. The last level consists of the specific indicators. These indicators define measurable or directly observable events. For example, in our problem one of the areas of concern is with troop movement. As shown under "Military -> Troop -> Deployment" branch, a specific indicator entitled "Move" is defined. We further define "Move" with the attributes of Division Name and Location. A second example is the indicator, "Threaten" under "Government -> Relationships with Leaders". It is worth noting here that we see the tool used by the analyst as a means to bring together the conceptual world (the way in which an analyst thinks) with the computational world (the way a computer works). As such the upper levels of the model define, in the view of an analyst the "problem" they are concerned with and their interrelationships. We note here that these interrelationships are simple and purely hierarchical. At the lowest level of our model are the indicators or actual events/observations. These indicators bind the conceptual and computational worlds together. It is envisioned that a library of indicators would be provided and the analyst would simply "attach" one or more indicators (possibly through a drag and drop) to a concept. It is these indicators that the model analysis techniques would be looking for. By separating the model in this manner, the underlying technologies used to implement the indicator(s) are hidden from the user.

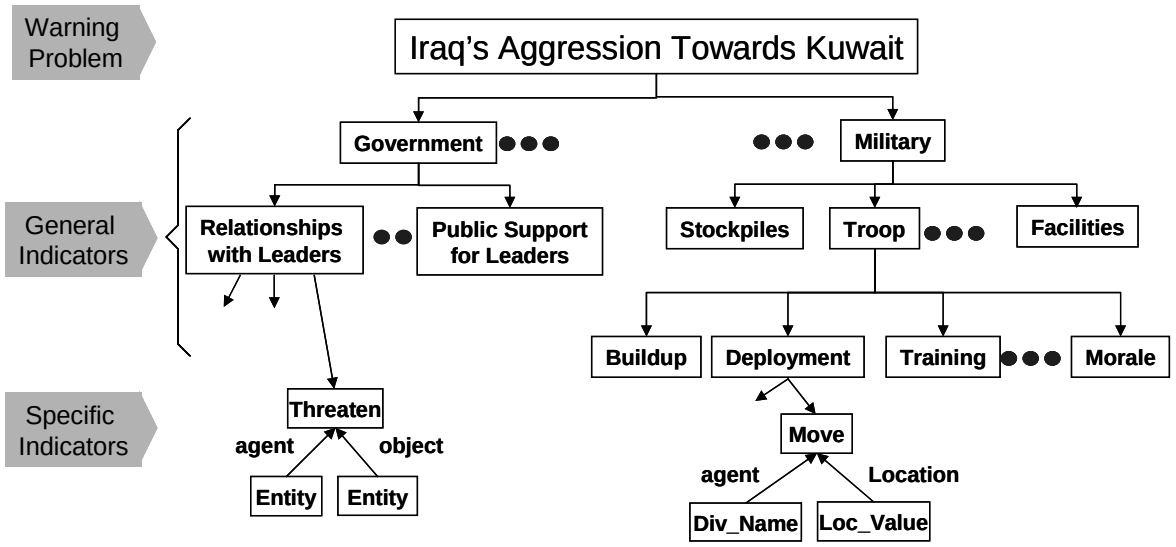

Fig. 5. Sample Warning Problem.

Once activated the model is stored in a model library for later use and is converted into a set of collection requirements for the data collector. In this case, the data collector is a product called Buddy Server. Buddy Server is a meta-search engine which can simultaneously query multiple sources for multiple requests (in the form of a topic tree). New documents, (not previously returned) are gathered and published on the JBI for downstream processing. Buddy Server performs the initial collection and schedules the requests on a regular basis to update the system. Buddy retrieves the document, and wraps the document with metadata. The metadata consists of a unique document ID for accountability, the keywords that retrieved the document from the source, the source's name, the date the document was retrieved, and the format of the document. This metadata allows downstream components to subscribe based on their capabilities and the document content.

Based on the format of the document, it is routed to the appropriate component. For the purpose of this demonstration we had both message-like traffic (e.g. formatted messages such as Tactical Reports) and free-text documents. The messages were routed to a system called the Generic Intelligence Processor (GIP) [4], while the free-text was routed to either Syracuse University's eQuery [6] or Cymfony's InfoXtract [9] parsers. The evidence database was then updated as each event was extracted.

On a periodic basis, a graph matching algorithm is run. The specific product used was 21st Century Technologies', "Terrorist Modus Operandi Detection System" (TMODS) [7]. The TMODS application periodically builds an input graph based on the evidence database and searches for subgraph isomorphisms of the target graph. Matches, either exact or inexact, are identified and those above a specific threshold are published. Based on the published results, alerts are brought to the analyst's attention through color changes on the original graph.

At this point the analyst can click on the indicator to see what events have been matched. The analyst can also bring up the original document in which the given event appeared. Figure 6 shows an architectural diagram of the components.

The components described in the aforementioned paragraphs provide us with an initial set of capabilities. It was intended to be small in scale and simplistic in order to provide a starting point. It is our goal to extend these capabilities by adding additional functionality, other components, a larger and more comprehensive data scenario and the implementation of a set of metrics.

3.2 Metrics

The initial efforts described in this paper were aimed at validating the Situation Awareness Framework and proving that the identified components could work together. With the integration now complete the ultimate goal is to establish an accurate measure of the system's performance and effectiveness. The success of any Situation Awareness system depends upon understandable Measures of Performance (MOP) and Measures of Effectiveness (MOE). These measures must include both quantitative and qualitative characterizations and be directly tied to the mission of the system in question.

At an abstract level the system may be viewed as a black box classifier. As such, the system may be evaluated in a similar manner with metrics such as precision, recall, area under the ROC curve, etc… However, the difficulty arises in understanding these results. In order to accurately characterize the system, one must have a technique to characterize the input to the system. Such a technique must not only capture the differences between various test datasets, but also

between test datasets and the real world. Ideally, these measures should also be independent of the technological approaches within the system. Initial attempts at these tasks have drawn on graph theory and the notion of signal to noise ratios often used in signal processing. The realization of these metrics would not only serve to evaluate existing systems, but also provide a true measure of progress.

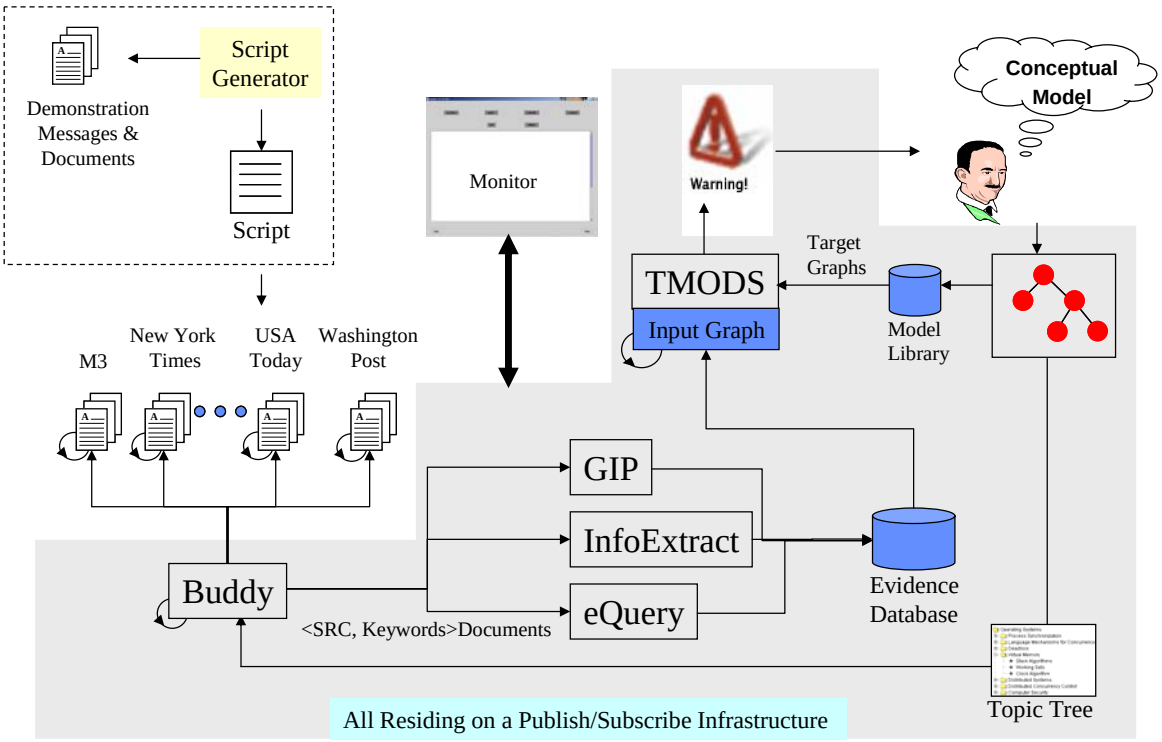

Fig. 6. Demonstration Architecture

4 Conclusions

Today, Situation Awareness is focused on the tactical picture and is reactive, instead of strategic and pre-emptive. Research under the higher levels of fusion will enable rapid understanding of strategic intent and impact assessment by future strategic planners and thus support Information Dominance. In this paper we have presented an initial framework for acquiring Situation Awareness. What is presented here is only a starting point. Work will continue to bring components together and to use this process to validate our overall conceptual model.

Acknowledgements

The authors would like to thank Robert Labuz and Dave Barnum of R&D Associates for many long hours of discussions and assistance in the implementation of the functional demonstration.

References

[1] Saso Dzeroski. Multi-Relational Data Mining: An Introduction. In *Newsletter of the ACM Special Interest Group on Knowldege Discovery and Data Mining*, Vol 5(1), July 2003.

[2] Mica R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors Journal*, Volume 37(1), pages 32-64, March 1995.

[3] Mica R. Endsley. Theoretical underpinnings of Situation Awareness: A Critical Review. Mica. R. Endsley, and D. J. Garland (editors), In *Situation Awareness Analysis and Measurement* (pp. 3-32). Mahwah, NJ: Lawrence Erlbaum Associates Inc.

[4] Walt Gadz,., Anthony Colby, and Michael Seakan. Information Extraction for Counterdrug Applications. In *2003 Proc of the ONDCP International Technology Symposium*, San Diego, CA, July 2003.

[5] David Jensen. Statistical Challenges to Inductive Inference in Linked Data. *Preliminary Papers of the Seventh International Workshop on Artificial Intelligence and Statistics*, 1999.

[6] Nancy McCracken. Representing Textual Content in a Generic Extraction. *2002 AAAI Symposium, Acquiring (and Using) Linguistic (and World) for Information Access*, http://cnlp.org/publications/SSS102NMcCracken.rtf.

[7] Sherry Marcus and Thayne Coffman. Dynamic Classification of Suspicious Groups Using Social Network Analysis and HMMs. In *Proc of the 2004 IEEE Aerospace Conference*, March 6-13, 2004.

[8] Barry McGuinness and Louise Foy, A subjective measure of SA: The Crew Awareness Rating Scale (CARS). In *Proc of the First Human Performance, Situation Awareness, and Automation Conference*, Savannah, Georgia, October 2000.

[9] Rohini K. Srihari, Wei Li, Cheng Niu and Thomas Corne. InfoXtract: A Customizable Intermediate Level Information Extraction Engine. *Workshop on the Software Engineering and Architecture of Language Technology Systems (SEALTS), North American Chapter of the Association of Computational Linguistics-Human Language Technology Conference 2003* (NAACL-HLT 2003), Edmonton, Alberta, Canada

[10] Alan N. Steinburg, Christopher L. Bowman, and Franklin E. White. Revisions to the JDL Data Fusion Model, *presented at the Joint NATO/IRIS Confer*ence, Quebec, October 1998.

[11] SWISH-E available at http://swish-e.org/.

[12] Ian Witten, Eibe Frank, and Jim Gray, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufman Publishers, Oct 1999.

[13] Douglas Holzhauer1, Vaughn Combs, Mark Linderman, Robert Duncomb, Jason Quigley, Mark Dyson, Robert Paragi, David A.Young, Digendra Das, and Carrie Kindler. Building an Experimental Joint Battlespace Infosphere (YJBI-CB). http://www.dodccrp.org/6thICCRTS/Cd/Tracks/Papers/Track5/093_tr5.pdf.

[14] Zhongfei Zhang, John Salerno, Maureen Regan, and Debra Cutler. Using Data Mining Techniques for Building Fusion Models. In *Proc of SPIE: Data Mining and Knowledge Discovery: Theory, Tools, and Technology V*, Orlando, FL, Apr 2003, pp. 174-180.