ALARM for Early Warning: A Lightweight Analysis for Recognition of Menace

Kellyn Rein
FKIE ITF
Fraunhofer
Wachtberg-Werthhoven, Germany
kellyn.rein@fkie.fraunhofer.de

Miłosław Frey
FKIE ITF
Fraunhofer
Wachtberg-Werthhoven, Germany
miloslaw.frey@fkie.fraunhofer.de

Ulrich Schade
FKIE ITF
Fraunhofer
Wachtberg-Werthhoven, Germany
ulrich.schade@fkie.fraunhofer.de

Silverius Kawaletz
FKIE ITF
Fraunhofer
Wachtberg-Werthhoven, Germany
silverius.kawaletz@fkie.fraunhofer.de

Abstract – Information overflow is a significant and as yet unresolved problem for military, homeland security and law enforcement. Furthermore, quantity is only one part of the intelligence problem: identifying which pieces of information belong together, and assessing the credibility of not only individual pieces of information but also of their correlations to one another complicates things. Automatically sifting, sorting and fusing information garnered from multiple sources into recognizable patterns of behavior and potential threats would provide a distinct operational advantage. This advantage would be clearly increased if the time needed for processing was close to real-time. A number of systems for deeper analysis of potential threats using technologies such as Bayesian networks exist, but tend to be time-intensive. This paper describes a near real-time solution for first-pass processing of inflowing information to provide early warning of developing threats.

Keywords: threat recognition, information fusion, situation analysis.

1 Introduction

Today's asymmetric military and security operations rely more upon information than upon firepower. Knowledge is definitely power[1]. The enemy no longer wears uniforms and marches in formation, but rather hides around corners or in plain sight. Success in operations requires the ability to quickly and appropriately discern patterns in and interpret the meanings of information which floods in from a variety of sources. Automatically sifting, sorting and fusing information garnered from multiple sources into recognizable patterns of behavior and potential threats would provide a distinct operational advantage. A number of systems for deeper analysis of potential threats using technologies such as Bayesian networks exist, but tend to be time-intensive. This information advantage would be clearly increased if the time needed for processing was close to real-time.

This paper describes a proposal for a linear-time solution – ALARM "A Lightweight Analysis for Recognition of Menace" – for first-pass processing of inflowing information to provide early warning of potential developing threats. This concept expands upon previous work [2][3], and at the same time resolving a remaining open problem, that of constraints. The solution is composed of three basic phases:

Phase 1 : Conversion. Parsing of incoming (natural language) information and conversion to a standardized language format for automatic processing, in our case Battle Management Language (BML). As part of the conversion process, each BML statement is assigned a value associated with its estimated reliability, based upon the perceived credibility of both the source of the information and the information itself, adjusted for uncertain formulations such as modal expressions.

Phase 2 : Checklists. After conversion, the incoming information is filtered via lookup tables for potential matches to (previously constructed) "checklist" models for correlating individual pieces of information. Fusion of filtered information into relevant model instances, based solely upon correlation with existing elements in the instance, generating a cumulative assessment of credibility, reliability and evidential weight within each instance. Once a given instance exceeds a (pre-assigned) threshold of evidential weight at phase 2, the thus presorted information contained therein is passed on to phase 3.

Phase 3: Process-based analysis. At this phase, the clustered information is examined more closely for specific constraints (e.g., time sequencing) not defined in phase 3. The threshold value of the cluster is modified (increased, decreased or cancelled) based upon the analysis at this phase. Depending upon the results of this analysis, the cluster of information may be forwarded to an analyst for further processing

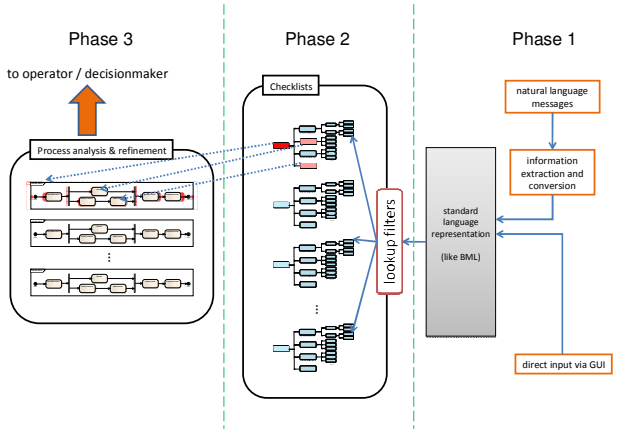

Figure 1. Overview of ALARM, showing the three phases.

Each of these phases will be discussed in more detail in the paper below.

The paper is organized as follows. Section 2 describes the underlying principles for the first phase, including the reasons for conversion of natural language information into a standardized form, a description of Battle Management Language as a standardized language, and a brief description about the conversion process itself. Section 3 describes the construction of the "checklists" and lookup tables for the filtering and presorting process in phase 2. Section 4 describes process analysis phase in more depth. And finally, in Section 5 we will discuss conclusions and recommendations.

2 Phase 1 - Language

Information which has been obtained from a variety of sources is generally in a variety of different formats. This can pose a significant hurdle for automatic fusion of the different pieces of information. Converting available information into a standardized format would greatly support fusion.

2.1 Conversion for processability

In ALARM conversion to a standardized language is necessary to not only create the lookup tables needed for the second phase, but also to simplify the mapping of individual pieces of information into the checklists in the second phase of processing. The standardized language which we are using is Battle Management Language (BML).

Additionally, as we are able to convert information from other natural languages into the single standard (see section 2.3), the system would support multinational endeavors.

2.2 BML as a basis for fusion

Originally designed for commanding simulated units, BML is a standardized language for military communication (orders, requests and reports) which has been developed under the aegis of the NATO MSG-048 "Coalition BML" and has been expanded to communicate not only orders but also requests and reports. BML is based upon the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) which is used by all participating NATO partners. As NATO standard, (STANAG 5525), JC3IEDM defines terms for elements of military operations, whether wartime or non-war, and is sufficiently expressive to formulate both military and non-military communications for a variety of different deployment types. It also provides a basis for standardized reporting among NATO coalition partners.

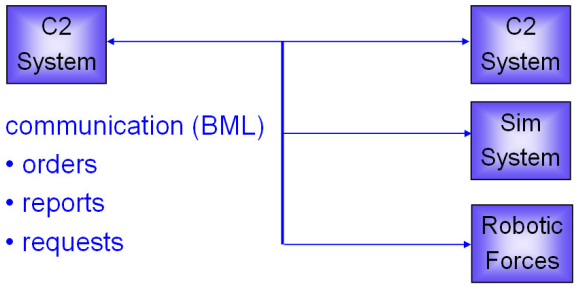

Figure 2. BML is a formal language for military communications such as orders, reports, and requests, which provides a common format for information

BML has been designed as a controlled language[4] based on a formal grammar[5]. This grammar is modeled after one of the most prominent grammars from the field of computational linguistics, Lexical Functional Grammar (LFG) [6]. This renders BML an unambiguous language which can easily be processed automatically.

Of particular interest is that BML statements may be represented by feature-value matrices, allowing fusion of individual communications through unification, a standard algorithm in computational linguistics [7]. Since data retrieved from databases and ontologies may also be easily represented as feature-value pairs, BML structure facilitates the fusion of not only field reports from deployed soldiers and intelligence sources, it also supports fusion of these reports with previous information stored as background information.

As described in [8], a basic report in BML is a single (atomic) statement which delivers a "fact" about an

individual task, event or status. A task report is about a military action either observed or undertaken. An event report contains information on non-military, "non-perpetrator" occurrences such as flooding, earthquake, political demonstrations or traffic accidents. Event reports may be important background information for a particular threat: for example, a traffic accident may be the precursor of an IED detonation. Status reports provide information on personnel, materiel, facilities, etc., whether own, enemy or civilian, such as number of injured, amount of ammunition available, condition of an airfield or bridge.

2.3 Conversion to BML

We have previously presented a method to analyse HUMINT reports written in natural language [9]. Our process of pre-analyzing natural language reports starts with information extraction (IE) based on the work of Hecking who applied IE techniques to the analysis of battlefield and HUMINT reports [10]. For information extraction, we use the freely available open-source tool GATE[11] where we run our data through the standard IE processing pipeline. This pipeline consists of the following elements:

1. A tokenizer that determines individual tokens of the text, i.e. single words, numbers, abbreviations and punctuation marks.
2. A gazetteer that compares the tokens to elements of several lists which contain names of various types. There are usually lists for person names, organisations, countries, places, villages and the like. Tokens matching one or more elements in the list will be annotated with the respective type, e.g. *female forename*.
3. The sentence splitter determines the boundaries of sentences, which is less trivial than it may seem at first glance. A certain built-in intelligence is required to prevent the sentence splitter from suspecting the end of a sentence after every period. Without it, a sentence would never make it past a "Mr." or "Dr." or any other abbreviation of that kind.
4. The Part-of-Speech-Tagger that comes shipped with GATE is a rule-based tagger with a lexicon under the hood. The tagger determines the part-of-speech of the word tokens according to the categories of the Penn-Treebank tagset [12]
5. A named-entities Transducer combines elements annotated by the gazetteers in step 2 above. For example, for the sequence "Dr. Mohammed el-Baradei", the gazetteer will provide the annotations *title* for "Dr.", *male forename* for "Mohammed" and *surname* for "el-Baradei" whereas a named-entity transducer uses these annotations to calculate the annotation *person* for the whole sequence.

The method uses shallow information extraction techniques based on GATE. We alleviate the disadvantages of the shallow approach by using ontological knowledge about verbs and their frames. The verbs and frames we consider are taken from the HUMINT domain. The frame information attached to a verb constrains the semantic roles that can be assigned to the sentence's constituents.

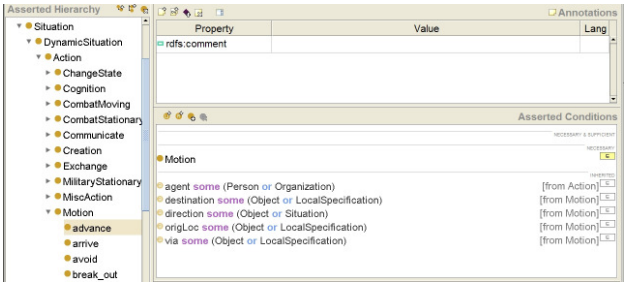

Figure 3: This snippet from a Protégé screen shows the semantic properties of the verb *advance*.

The method presented for report analysis can be a component of larger systems, e.g. machine translation systems that translate reports into all languages being used in a complex combined operation. For example, at present we are working on a prototype which converts reports from German into BML for further processing.

2.4 Credibility assignment

There are several important elements to BML basic reports for the fusion process. First is the fact that each BML "report" is a statement representing a single (atomic) statement. Second is that each basic report has its own values representing source and content reliability. Third is that each report also has a reference label to its origination so that the context is maintained for later use by an analyst.

The first point (atomicity) is essential for the fusion process: each statement of a more complex report may be processed individually.

However, this atomicity is additionally significant for the second point (uncertainty evaluation). Natural language text sources such as HUMINT reports usually contain multiple statements. Some of these statements may be declarative ("three men on foot heading toward the village"), other statements may be speculative ("possibly armed"). While an analyst may assign a complex HUMINT communication an overall rating (e.g., using the familiar "A1"-"F6" system), individual statements contained therein have greater or lesser credibility. Therefore the conversion to BML assigns first the global rating, but adjusts each individual statement according to the uncertainty in its formulation, e.g., on the basis of modality term analysis.

3 Phase 2 – Checklists

In general, soldiers and intelligence analysts have mental checklists ("rules of thumb") of events or conditions that they watch out for as signs of potential developing threats or situations. For example, the checklist of the factors which might constitute forewarning of a potential bomb attack on a camp would include such things as the camp appearing to be under surveillance, reports that a local militant group may have acquired bomb materiel, and a direct tip from an informant concerning an attack. Many of these factors may be further broken down into more detail, the matching of which "triggers" the activation of the factor. For example, the acquisition of blasting caps would activate the factor "bomb materiel". We can represent the hierarchy defined within the checklist as a simple tree-like structure as shown in Figure 4.

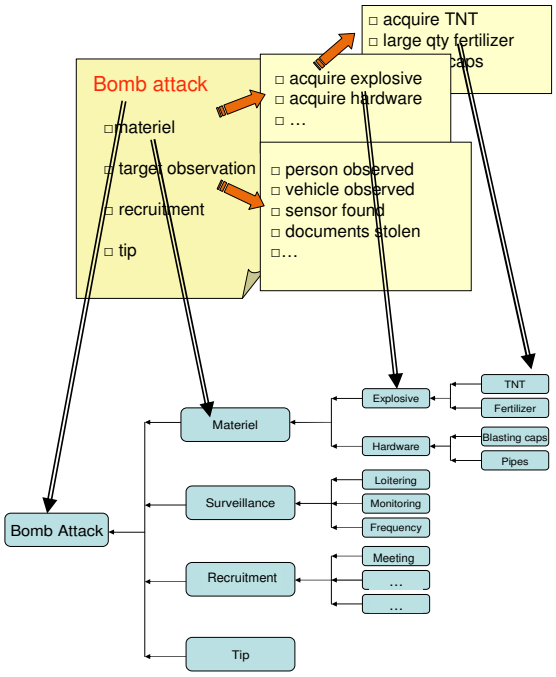

Figure 4. Checklist converted to a tree structure

When new information flows in, a human analyst will try to find if this information is relevant for her purposes. She will attempt to sort and correlate the various pieces in order to try to construct a coherent picture. In this phase of ALARM we rely on two mechanisms for this:

1. Each piece of information which triggers a specific element of one or more checklists is contained in a lookup table, which indicates which threats it triggers; a,
2. We define within the checklist model itself how various elements must be related in order to be relevant.

The trigger lookup table contains specific patterns (usually a BML verb plus one or more elements of the BML statement which must match). For example, using the checklist appearing in Figure 4, the lookup table would contain the BML entries "procure TNT", "procure fertilizer", and "procure blasting caps". In this case, the conversion from Phase 1 would have standardized various natural language forms indicating "gained possession in some method" (i.e, bought, stole, etc.) to "procure." If we receive a message with "procure goat" and we are not interested in tracking goats, the statement will be ignored.

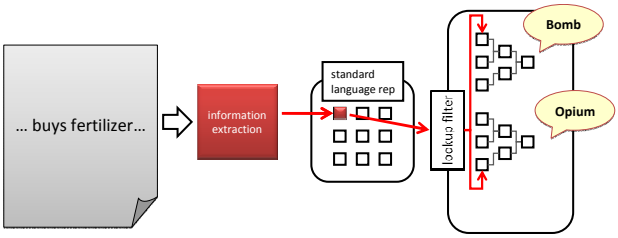

Figure 5. Mapping new information into the model.

When a pattern is matched to an entry in the lookup filter, it will be passed to one or more checklists for further processing. An example of multiple matching would be "procure fertilizer": in our area of activity large quantities of fertilizer may indicate threat of a homemade bomb (a direct threat) or the cultivation of opium (an indirect threat). Weights according to the (heuristically determined) relative frequency of each are built into each model, i.e., fertilizer is a weaker indicator of a bomb than of opium cultivation (for a more in-depth discussion see [13][14].

If there are no existing instances of the threats, a new instance will be created. However, should there already be one or more instances, our new information will be checked for relevance to the existing information based upon how the various pieces of information relate to one another. For example, in Figure 6 we can see that only trigger concerning a specific location will be added to the "surveillance" cluster. However, as the enemy will not build the bomb intended for the location at the location itself, we connected "surveillance" and "bomb" by connecting individuals or organizations.

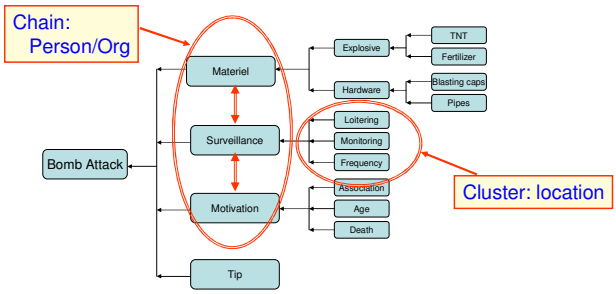

Figure 6. Representation of correlations in threat model.

Within a given structure different elements are weighted as to how significant an indicator of the threat they are (local evidence, i.e., significance within a given threat structure). For example, while "fertilizer" may be a trigger for bomb materiel, it may not be as strong an

indicator as, say, “commercial explosive”, would therefore have a relatively low weighting.

The various weights -- the credibility (source, content) of the initial information, the evidential weighting between and within models – interact to assure that there is a certain amount of checks and balances: unreliable information (assigned a low credibility) may trigger a strong indicator for a threat, but doubt is covered through the balance of the weights.

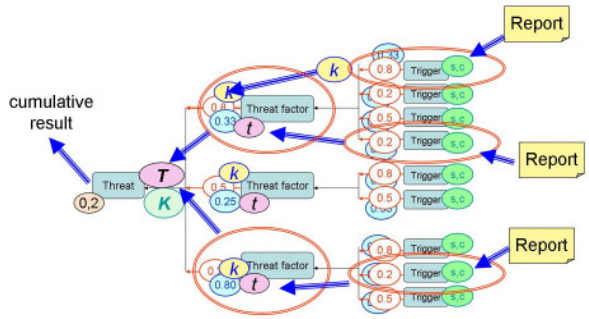

Figure 7. Flow of calculation through a checklist tree.

As more information flows into a model instance, the cumulative result increases and eventually reaches a predefined threshold, at which point the information contained in the checklist is passed on to the third phase for constraint and process analysis.

4 Phase 3 - Process-based Analysis

Gathering facts alone is not enough for a threat analysis. A system for analysis of potential threats has to be able to discover further dependencies and relationships between those facts and to match them against known behavioral patterns.

4.1 Overview

The checklist as described in the preceding section provides a mechanism for clustering possibly relevant information, but is unable to define more intricate relationships such as, for example, sequencing of events which may be pertinent to the identification of developing threats. For this task, we suggest a subsystem, which is based upon workflow analysis. The workflow description is derived from business process analysis and defined with a few elements like actions, gates and other items defining the information flow.

For efficiency of operation, this subsystem would only be activated when a cluster of related information is forwarded on from the previous phase, i.e., there appears to be sufficient information in the cluster of related information to make further analysis worthwhile.

4.2 Networks & Tokens

The central part of workflow analysis consists of a network with active nodes. Networks have to be built by analysts, which have enough domain knowledge to reproduce all possible dangerous processes in a generic manner.

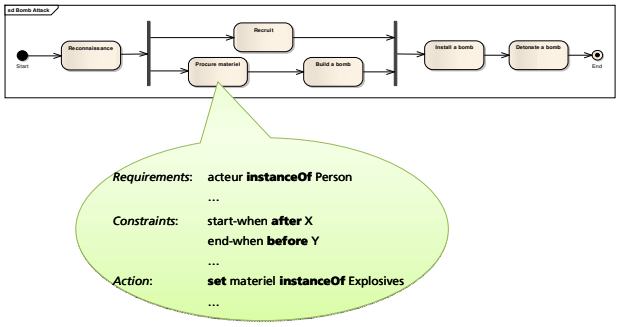

Figure 8. A process node describing constraints.

Nodes depict activities. The central part of an activity is a verb including its semantic frame. Additionally, more constraints are to be defined. The first one is a list of prerequisites. These are logical expressions matched against incoming information. Prerequisites define the minimal state of the network required for a given node to be activated. Further constraints describe temporal and spatial dependencies within the incoming data. The last part of a network node’s definition is the set of actions which are to be performed when a node becomes active. The actions manipulate information carried by a token and send them to next nodes, if applicable.

A token defines a piece of information sent through a network. Its meaning is twofold. At first it is just propagated through network, carries information and activates nodes. Secondly, its position in network defines how far a possibly dangerous action is advanced (see below).

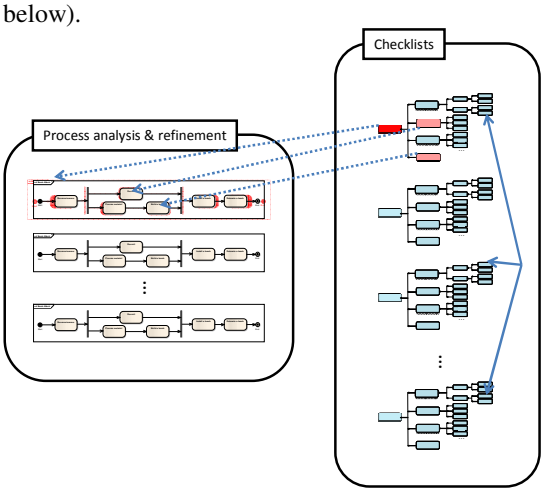

Figure 9. Mapping of active checklist elements to corresponding process.

The active elements of the checklist instance are mapped into a second model which examines in more depth dependencies and relationships (e.g., time sequencing, etc.) necessary for accurate assessment of the situation modeled.

All processes in the system represent potentially dangerous activities. These processes have been defined by domain experts and are activated based upon the information matches found in the scanning phase. The information from checklists initializes matched actions, which are parts of activated processes. The "activation value" defines how far a possibly dangerous activity from its completion is.

The process analysis consists of the following steps:

- The system creates a token, which is one of the main artifacts in this part of analysis.
- A token holds information about the completeness of a given process and is passed by a completed action to its followers.
- An action decides, based on the information in received token, whether all prerequisites are met and whether there is enough information available to proceed.
- A token holds also an "activation value" which indicates a completeness of a process. This value is modified by actions. This activation value also indicates the threat level

According to the definition of a process, the system propagates tokens from one action to another. Where the dependency analysis fails, the instances can be discarded (ignored), when the dependency analysis holds, the results will be forwarded to an analyst for decision-making and action.

4.3 Threat Level

One of the most important things in the process-based analysis of threat is that no threat model as such is built or used. The analyst builds simply a model of a process which leads possibly to a dangerous situation.

The threat itself is calculated during the activation flow through the network. The initial threat value from the checklist is passed on to this phase, and is modified (increase or decreased) during this analysis phase. As mentioned previously, a token is passed though the network. The token also carries, besides other information, an "activation value". The threat level is measured by the "activation value".

The activation value is modified by network nodes based on the prerequisites met and other constraints, as well as on an individual value of a node. The individual node value describes the importance of this node for the whole process.

When the dependency analysis succeeds, and returns a threat level which is considered significant, the information cluster, as well as details of the analysis will be forwarded on to analysts and decision-makers for further review.

5 Summary and Conclusion

Figure 10 shows a complete overview of the flow through the proposed ALARM system.

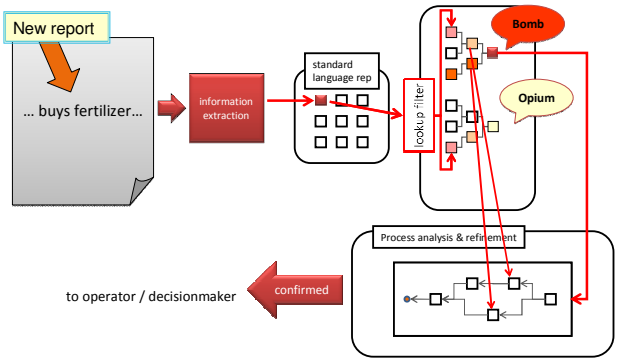

Figure 10. Flow of information through ALARM system.

As mentioned earlier in this paper, and as is apparent from the word "lightweight" in the name, ALARM is not intended to replace deeper and more subtle methodologies for analysis to detect new, previously unknown relationships between information. Rather, ALARM has been conceived to provide rapid, first pass warning of potential developing threats, based upon heuristic knowledge of the area of operations and upon observed behavior of the enemy.

Many of the parts of the system as described above have been or are currently being implemented (e.g., natural language parsing and conversion module), others are still in the design phase.

We believe that this system can provide an interesting complement to existing systems and will prove to be a useful tool.

References

[1] D.S. Alberts and R.E. Hayes. *Power to the Edge*. Washington, DC: CCRP, 2003.

[2] K. Rein and Frey, M. "Towards a Simple Model for Predictive Threat Assessment." MCC 2009, Prague, Sept. 2009.

[3] K. Rein, A Simple Heuristics-Based Model for Threat Prediction With Uncertainty Evaluation to Support Decision-Making. CCRP Journal (submitted), 2010.

[4] W.-O. Huijsen. "Controlled Language – An Introduction," in *Proc. of the Second International Workshop on Controlled Language Applications (CLAW98)*. Pittsburgh, PA: Language Technologies Institute, Carnegie Mellon University, May 1998, pp. 1-15.

[5] U. Schade and M.R. Hieb, "Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports." *11th ICCRTS*. Cambridge, UK, 2006.

[6] J. Bresnan. *Lexical-Functional Syntax*. Malden, MA: Blackwell, 2001.

[7] S.M. Shieber. *An Introduction to Unification-Based Approaches to Grammar* (= Volume 4 of CSLI Lecture Notes Series). Stanford, CA: Center for the Study of Language and Information, 1987.

[8] U. Schade and M.R. Hieb, "Battle Management Language: A Grammar for Specifying Reports." *2007 Spring Simulation Interoperability Workshop* (Paper 07S-SIW-036). Norfolk, VA, Mar. 2007.

[9] C. Jenge, S. Kawaletz and U. Schade. "Combining Different NLP Methods for HUMINT Report Analysis." NATO RTO IST Panel Symposium, Stockholm, Sweden , October, 2009.

[10] Matthias Hecking. Information Extraction from Battlefield Reports. In *Proceedings of the 8th International Command and Control Research and Technology Symposium (ICCRTS)*, Washington, DC, U.S.A., 2003.

[11] Gate: A general architecture for text engineering. http://gate.ac.uk/.

[12] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313 – 330, 1993.

[13] K. Kruger, U. Schade and J. Ziegler. "Uncertainty in the fusion of information from multiple diverse sources for situation awareness." *Proceedings Fusion 2008*, Cologne, Germany, July 2008

[14] K. Kruger. "Two 'Maybes', One 'Probably' and One 'Confirmed' Equals What? Evaluating Uncertainty in Information Fusion for Threat Recognition," *Proceedings MCC2008*, Cracow, Poland, Sept. 2008.