Towards Automatic Threat Recognition

Ulrich Schade, Joachim Biermann, Miłosław Frey
FGAN-FKIE
Neuenahrer Straße 20
53343 Wachtberg-Werthhoven
GERMANY

{schade|j.biermann|m.frey}@fgan.de

ABSTRACT

The current transformation processes aim at robustly networked forces in order to enable them to execute network-centric operations [1]. Obviously, this will provide the forces' headquarters with a huge amount of data and information that has to be processed to deduce an appropriate picture of the respective battlespace and evolving threats in a timely and most reliable manner. Some of the data will not be relevant at all, but some other may indicate upcoming threats. This is true especially with respect to reconnaissance reports which will not only include information from own reconnaissance assets but also from civilians and other open sources. Thus, one of the most challenging aspects of data and information processing in military affairs is to find the situation relevant information within the huge amount of irrelevant one.

The paper at hand describes a high level information fusion system under development. This system automatically processes the incoming stream of unstructured information in order to support the human operator in intelligence processing. Ideally, analysing the input information based on the so far perceived situation and available background knowledge, the system recognises that part of information which might indicate new threats (as well as targets) or confirm existing hypotheses. The relevant and pre-processed information then will be presented to the operator for further (interactive) investigation.

Our system processes the incoming messages in two steps according to the established intelligence processing procedures. During the first one, the reported events are categorised (or classified), and cross-referenced. In the second one, the resulting information is analysed and combined and integrated into patterns in the course of the production of further intelligence in order to separate the presumable relevant information from the less significant one. For both steps, knowledge about the domain and the information context has to be accessed and exploited. Therefore, the system resorts to an ontology. To point out the system and its functionalities, the paper will describe this ontology as well as the processing steps in general and by example.

1.0 INTRODUCTION

To avoid information overload, the input of a C2-system should be pre-processed automatically to support the human operators. One of a C2-system's modules for the automatic pre-processing of incoming data and information should be a system for high level information fusion which processes the incoming information in order to identify indications of threats (and targets as well).

In contrast to sensor data fusion which identifies objects by applying mathematical methods on underlying physical models, high level information fusion either needs behavioural models describing the actions

Schade, U.; Biermann, J.; Frey, M. (2006) Towards Automatic Threat Recognition. In *Information Fusion for Command Support* (pp. 11-1 – 11-10). Meeting Proceedings RTO-MP-IST-055, Paper 11. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

other factions of interest might undertake or needs normalcy models describing common and unsuspicious behaviour (for a more detailed discussion about the differences between sensor data fusion and high level information fusion, cf. [8, 3, 5]). Behavioural models, on the one hand, are needed for identifying the behaviour of other factions of interest out of some parts of the behavioural patterns which occur in the incoming data. For instance, eight hours after initiating the ground campaign ("Desert Storm"), General Schwarzkopf got the information that the Iraqis had destroyed the desalination plant of Kuwait City. He, then, rightfully inferred from this partial information the operational behaviour of the Iraqis: they were pulling out of the city ([20], p. 453). Normalcy models, on the other hand, are needed for the detection of deviations from the normal. For instance, in the war against terrorism, a person entering a subway station on a hot summer day constitutes a suspicious deviation if this person is wearing a thick winter coat.

In the following, we will present aspects of a system for high level information fusion in development. We will describe its mode of operation in general and by example. The main example discusses the processing of the following report taken from [4]:

Report I/61: 091629A00SEPT by UN-MAC, group TESLIC, own position at TYK286415:

"6 TAM110 carrying soldiers passed at 1515 Teslic (TYK2642), direction Gusci (TYK2840), speedily. One vehicle carried boxes. The vehicles returned at 1600, empty."

2.0 THE SYSTEM

To be able to aggregate and fuse the information from a set of independent reports, a system has to exploit knowledge. Therefore, it depends on a module which provides the knowledge needed. This module is an ontology component. In order to explain the functionality of our system and the role of its ontology component, "ontology" is discussed under a more general view in the following subsection. Subsequent to this discussion, the operational structure of intelligence processing is referenced in order to explain which steps of this processing are covered by the system. At last, the system itself is presented, from explaining its basic processing principles to a discussion of its processing steps by example.

2.1 Ontology

Classically, "ontology" denotes the study of being as such (the science of being qua being). Recently, however, "ontology" entered the terminology of computer science. Here, "ontology" is defined as an explicit specification of a conceptualization [7, 19]. In short, an ontology represents within the machine (human) knowledge about a specific domain. In our case, the domain in question is the domain of military operations and intelligence in the context of conventional war operations as well as in the context of operations other than war (OOTW).

To build an ontology is a formalized way to represent knowledge, normally supported by an ontology editor which in our case is Protégé (http://protege.stanford.edu/). An ontology consists of three parts. First, there is a taxonomy about the objects relevant in the domain. Second, there is a set of attributes by which relevant features of these objects can be described. Each attribute comes along with a domain of values. Last, there is a set of relationships holding among objects, attributes, and relationships. Rules which cover objects' templates of behaviour or their activity patterns are formally represented as relationships (among relationships). For example, our ontology includes an object "Roland." According to the taxonomy, a "Roland" is an anti-aircraft weapon system which is a member of the object class "vehicle." As a member of "vehicle," "Roland" has the attribute "mobility." Its value is "land tracked." With respect to "land tracked," the ontology includes a rule, saying that land tracked as well as wheeled vehicles are self-propelled (in contrast to towed vehicles). In addition, there is a rule saying that self-propelled vehicles possess an attribute "default speed on roads" and that "land tracked" vehicles also have an attribute "cross-country default speed." Besides, there are more complex rules about terrain access which are based on the speed rules mentioned. Most advanced among these rules, in the case of the system presented in the paper at hand, are schemata. Schemata are stored in the ontology's scheme base. They describe military tasks as sequences of generalized events, and they are used as patterns against which sets of observed events are matched by the system in order to recognize ongoing activities and threats as will be exemplified in the following sections.

We already have developed an ontology for the military domain. This ontology has been tested within a C2 system in two ways. On the one hand, the ontology is part of the so-called "SOKRATES" prototype [16, 17] which analyzes reports written in natural language and displays the results (which are the contents of the respective reports) on a map. In this application, the report first is transformed into a formal representation by means of information extraction [9]. Then the ontology is used to enrich the formal representation semantically. Both steps together can cope with the vagueness and the ambiguity of natural language. On the other hand, the ontology is used in order to increase interoperability of information exchange between an army and an air force system. The information provided by the army system is transformed and whenever possible enriched to make it suitable as well as significant for the air force system. In particular, the exchange applies to information about movements and positions of friendly as well as of hostile ground forces. Air forces need to know the position of own troops and protected areas in order to avoid fratricide and accidental collateral damage, and they also need to know the enemy's position in order to deduce target locations and anti-aircraft threats. The ontology module calculates the entire area covered by own troops from reported positions and individual spatial operational areas. It also highlights hostile anti-aircraft systems. This is done during the information flow from the army to the air force. An adapted version of this ontology is now used within our system for high level information fusion.

2.2 The operational structure of intelligence processing

A high level information fusion system performs part of the tasks to be done in intelligence processing. Intelligence cells have to process and evaluate incoming information in order to develop a most accurate situational awareness of the battle space prior to all own decisions and activities. This is done in a structured and systematic series of operations which is called the Intelligence Cycle. It includes four stages *Direction - Collection - Processing - Dissemination* which are defined by the NATO Glossary of Terms and Definitions (AAP-6) [14]. These four discrete stages are conducted culminating in the distribution of the finished intelligence product [2].

Processing is where the information that has been collected in response to the direction of the commander is converted into intelligence products. It is a structural series of actions, defined as the production of intelligence through *Collation, Evaluation, Analysis, Integration,* and *Interpretation* of information and/or other intelligence. The system described in this paper focuses on a support of information exploitation and fusion within the Collation, Analysis, and Integration steps. The military understanding of these processing steps is kept in the Allied Joint Intelligence Counter Intelligence and Security Doctrine (AJP 2.0) [15] and in the NATO Glossary of Terms and Definitions (AAP6) [14].

In general, the incoming information consists of all kind of reports (from reconnaissance assets as well as from human sources) about observations of activities (e.g., movements of people, units, and vehicles), military sites, equipment, and others. It is the task of the intelligence processes to aggregate the categorized and correlated individual situation elements given by the incoming reports, based e.g. on available schemata, and to fuse them into more complex operational elements, by this creating new objects in the domain of concern (e.g., correlated march movements). In our systems, the schemata are stored in the ontology. Therefore, the comparison whether individual elements partially constitute and thus indicate an activity (the match of observed events against a scheme) is a corresponding ontological process.

2.3 Unification

Matching a set of events against an ontological scheme is done by means of unification ([18], [6]). Unification is the main processing principle within our system. Its use in particular causes that the matching is ruled by a general algorithm. The matching mechanism neither depends on the specific events to be matched nor on the scheme against which the matching is performed. This has the obvious advantage that the system itself is flexible and adaptable: Schemata from the ontological scheme base can be changed as required and new schemata can be added as well. In order to fully exploit this advantage, however, a scheme editor has to be integrated in the system such that operators can adapt the system to their requirements on the fly.

In principle, schemata represent types of situation elements. Either a sequence of events or two instantiated schemata can be matched against a scheme by means of unification. If a sequence of events is matched against a scheme, it is tested whether the sequence of events constitutes a hypothesis of the given type. If the unification is successful, the scheme becomes instantiated and, thus, represents a hypothesis. In contrast, if the unification fails, the sequence of events does not constitute a hypothesis of the given type. If two instantiated schemes (two hypotheses) are matched against the scheme in question, it is tested whether these two hypotheses can be fused to a single one. Here, failure of unification means that the two hypotheses cannot be fused and therefore have to be kept separately for further processing.

2.4 The first processing step: Information Collation by Categorization and Cross-referencing

The system described in this paper proceeds in two major processing steps. The first one supports Collation which is defined by [14, 15] as follows: "*The [...] decomposition [of the incoming information] into individual information items. These individual information items are subject to categorizing according to either predefined categories or to new identified categories adapted to the mission. The categorized information items are finally cross-referenced each one with the others*." Categorisation, in military terms, can be understood as a classification of situation elements. It is followed by cross-referencing which is a step of correlation of identified items merging their attributes.

The example given in the introduction reports two events, in both cases, the movement of six trucks. In the first event, the trucks were loaded and went in the direction of Gusci ("*6 TAM110 carrying soldiers passed at 1515 Teslic (TYK2642), direction Gusci (TYK2840), speedily. One vehicle carried boxes.*"), and in the second event, they came back empty ("*The vehicles returned at 1600, empty.*"). The events and their spatial and temporal progression match a deployment which is quite obvious to a human military mind. In the system, this match is found as follows. Initially, the report is parsed and the resulting items are matched against the taxonomy which is part of the system's ontological representation of the domain. This processing is in accordance to the decomposition within the Collation step. A formal representation results. This representation has the form of a feature-value matrix (cf. figure 1).

```
type: move
theme:       type:         TAM110
             count:        6
             cargo:        [unit,consumable_materiel]
path:        type:         town
             name:         Teslic
             coordinates:  TYK2642
             pass_time:    2000_09_09_1515
direction:   modifier:     towards
             type:         town
             name:         Gusci
             coordinates:  TYK2840
speed: fast
```

```
type: move
theme:       type:         TAM110
             count:        6
             cargo:        empty

path:        type:         position
             coordinates:  TYK286415
             pass_time:    2000_09_09_1600

direction:   modifier:     towards
             type:         town
             name:         Teslic
             coordinates:  TYK2642

speed: normal
```

Figure 1: Feature-value matrices for the events reported as “6 TAM110 carrying soldiers passed at 1515 Teslic (TYK2642), direction Gusci (TYK2840), speedily. One vehicle carried boxes. The vehicles returned at 1600, empty.”

Feature-value matrices are not only a standard representation form in the field of computational linguistics, they also have at least two additional advantages. First, they can be easily notated in XML. Therefore, they can be exchanged between C2-systems which allows for at least level 2 interoperability [13]. Second, they can store *incomplete* information. In our system, the latter property is used to merge feature-value matrices resulting from different reports (cf. figure 2) and match them against behavioural schemata by means of unification. The deployment scheme in figure 3 describes the following behaviour: A military unit, denoted by the variable U, is moved from one place, denoted by A1, to another (A2) which it occupies. In the figure, temporal constraints are left out. The matrix of merged events shown in figure 2 can be matched to the deployment scheme matrix by means of unification.

```
list:  event: name:  I-61-a
              type:  move
              theme: type:       TAM110
                     count:      6
                     cargo:      [unit, ...]
                     ...

       event: name:  I-61-b
              type:  move
              theme: type:       TAM110
                                 count: 6
                                 cargo: empty
                     ...
```

Figure 2: The matrix of merged events shown in parts (without spatial and time information)

A successful match means that the sequence of events merged into the matrix is compatible with the respective scheme and thus indicates the scheme's activity.

```
list:   event: type:        move
               theme:       type:  X
                            count: N
                            cargo: [U|_]
               source:      A1
               destination: A2

        event: type:  occupy
               agent:U
               loc:    A2

        event: type:        move
               theme:       type:  X
                            count: N
                            cargo: empty
               source:      A2
               destination: A1
```

Figure 3: The figure shows parts of the deployment scheme.

As has already been stated, unification, in principle, is well-defined with respect to feature-value matrices. These matrices are sets of feature-value pairs and the value of a feature either is atomic or a feature-value matrix by itself. However, there is one restriction: A feature can only have one value. Thus, with respect to sets, to unify two matrices means to build the union of the features and assign a value to each feature. If a feature had been given a value only in one of the matrices, this value is kept. If it had been assigned values in both matrices, these values must be unified. Since an atom only unifies with itself, the unification of the values may fail. In this case the whole unification also fails.

With respect to the matrices and schemata which we use in our system, the unification has to be a little bit less strict. In particular, we allow feature values to be variables or lists besides being atoms or matrices. For example, in the scheme given in figure 3, there is the variable X. It represents the type of what is moving (move's theme). Besides, the matrices in figures 2 and 3 are lists of events. Lists permit to use a feature, here the feature "event," more than once in a matrix. Nevertheless, the strength and advantage of the unification mechanism is not affected. The order of a list's elements can be used to restrict the possibilities of unification. The same holds with respect to variables. In principle, any variable unifies to everything. Thus, the unification of the first events in the matrices shown in figures 2 and 3 is successful. However, this unification binds the variables involved. X is instantiated with "TAM110", N with "6", and U with "unit." As these variables hold within the whole matrix, the respective instantiations also hold for the third event of figure 3 and restrict unification.

The unification algorithm used is also less strict with respect to atoms. Whether atoms only unify to themselves depends on the type of the atoms involved. In particular, if an atom is a value of the feature "type," it denotes the class of an object in the ontology's taxonomy. In this case, if one of the values to be unified is a hyponym of another, unification succeeds with the hyponym as result. For example, unifying "leopard2A5" with "battle tank" results in "leopard2A5," but "leopard2A5" will not unify with "howitzer" because no relation of hyponymy holds.

Another example is the matching of moving events to one move scheme. Beside temporal and spatial constraints, a sequence of merged moving events can only match (and thus be fused) to a single move if the events' agents (the moving units or persons) as well as their themes (the moving vehicles) can also be merged. For example, if there is a report about moving battle tanks, a report about moving T72s and a report about a moving amoured unit, these reports can be fused if the temporal and spatial constraints allow it. In contrast, a report about moving battle tanks can not be fused with the reports about moving TAM110s given above: The themes of the moves, on the one hand the battle tanks and on the other hand the TAM110s, cannot be merged by unification because no hyponymy relation holds between "battle tank" and "TAM110." This fusion rule about moves is an ontological rule connected to the move scheme which is as all schemata stored in the ontology as well. The rule also ensures that the fusion result from our main example would also be valid if the reporting unit had mentioned the movement of 6 TAM110 in two independent reports without asserting their referential identity.

The result of the system's first step of processing is a set of activities which are compatible with some sequences of single events reported. The activities are represented in the form of matrices. The matrices result from unification processes merging single event matrices and the respective activity's scheme matrix. The resulting matrices still are incomplete. Their further completion is handled during the system's second processing step.

2.5 The second processing step: Analysis and Integration

The second processing step supports analysis and integration. The NATO definition of these steps are as follows:

Analysis: "A step in which information is subjected to review in order to identify significant facts for subsequent integration." Analysis, thus, consists of a number of interacting sub-processes to bring answers to questions like 'Who is it?,' 'What is it?,' etc. in order to set the stage for integration.

Integration: "A step whereby analysed information or intelligence is selected and combined into a pattern in the course of the production of further intelligence." Integration, thus, is the process of building pictures of the current situation providing by this a basis for the development of predictive situations. This is done to answer questions like 'What does it mean?,' 'Why is it happening (intention)?,' etc. in order to finally recognize threat indicators.

During analysis, the focus is set on those activity patterns which are compatible with the sequence of reported events, i.e., the patterns which result from the first processing step of information collation. The knowledge provided by the ontology is exploited in order to augment the activity patterns.

With respect to our main example, during analysis, the explicit speed of the TAM110 reported is estimated from [speed: fast] and from the knowledge about a TAM110's default and maximum speed provided by the ontology. Then, using this speed and the reported time information, the radius of operation is calculated. Therefore, geographical knowledge is required, especially knowledge about the road network in the area in question. Next, information about relevant locations, e.g., those mentioned in the reports as well as those estimated as source or destination of movements, is added. Finally, the unit determination process presented in [10, 16, 17] is triggered in order to provide an estimation about the size and the identity of the unit which executes the operation under analysis. The augmented matrices of the activity patterns are delivered to integration.

The core of the integration process is "intention." Every activity matrix is augmented by an intention feature, and the values of these features are estimated by integration. In principle, these values provide an answer to the question WHY the activity is undertaken. The value is either a label or a pointer to another activity matrix. All labels are activity types taken from and defined by C2IEDM's table "action-task-activity-code" [12]. For our main example, the value of the "intention"-feature should represent a valid guess about the reasons for the deployment. Under consideration of the situation, the calculated radius of operation, and additional geographical knowledge, higher level activities are checked whether they might be the reason for the deployment under question. Lorenz [11] already argued that "ambush" is a potential value. Under solidified information, the value even can be substituted by a pointer to an "ambush" matrix which can store the additional information, e.g., information about the most likely affected activity by the ambush (adjacent vulnerabilities). Obviously, an ambush constitutes a threat. In this sense, the integration process provides threat indications.

Naturally, integration as well as the preceding processing substeps draw on knowledge, not only knowledge about military operations or geographical features, which dominate the ambush example, but also situational knowledge. This can be illustrated best by the simpler example of the movement. During the analysis substep the path of movement activities is analyzed. If such an analysis shows that the destination of a movement (say of a common car) might be an own area of engagement and its source is known as "location of adversarial forces," a possible intention of the move is a hostile act. From the processing point of view, a specific value is attributed to the intention feature if the activity in question matches (in terms of unification) a matrix which is the value of the presupposition feature of another activity. In this case, the type value of this more complex activity is taken as intention value for the activity in question. In the movement case, the intention can be a hostile act or to be more specific a terrorist act. Such kind of values indicate a threat, and, thus, the respective moving activity will be presented to the human operator for further investigation.

3.0 CONCLUSION

The paper at hand introduces a system under development. The system's task is automated support for threat recognition. Its development is determined by two major principles of importance. The first of these principles applies to modularity. In the system, modularity becomes manifest in the fact that all domain knowledge is stored in the ontology component. The ontology describes all objects, structures, and relations of the domain. All processing (matching) is done uniformly by unification. It is thus governed by a set of general domain independent rules. All the specifics are kept in the schemata against which the reported sequence of events is matched. These schemata represent known activity patterns and the heuristic procedures to process information in order to deduce intelligence. Because these patterns are uncoupled from processing, they can be changed as well as modified by the operator. Thus, the system can be adapted to unknown situations and new activity patterns, a necessary property in times of asymmetric warfare.

The second principle our system meets is the “human in the loop”-principle. The human operator not only can adapt the system by schema manipulation, the operator also has the power of decision. The system only generates hypotheses about possible threats and informs the operator. Then, the human operator evaluates these hypotheses, possibly after demanding for additional information about the deduction of the presented hypotheses. In principle, the human operator has the following alternatives to react on a presented possible threat: a) judging the possible threat as real, the operator can initiate an alert; b) judging the event as ambiguous, the operator can complete a request pre-generated by the system to task reconnaissance assets; or c) judging the event as less relevant, the operator can leave further processing to the system.

REFERENCES

[1] Alberts, D.S. (2002). *Information Age Transformation*. Washington, DC: CCRP.

[2] Biermann, J. (2003). A Knowledge based Approach to Information Fusion for the Support of Military Intelligence. *IST Symposium on Military Data and Information Fusion*. Prag.

[3] Biermann, J. (2004). Challenges in High Level Information Fusion. *The 7th International Conference on Information Fusion*. Stockholm.

[4] Biermann, J. & Lorenz, F.P. (2003). Abschlussbericht zum Forschungsvorhaben T/I11S/0A359/M0416 vom 18.5.2000 'Ein Experimentalsystem zur wissensbasierten Meldungsauswertung in Führungssystemen'. Wachtberg: FGAN-FKIE.

[5] Biermann, J., et. al. (2004). From Unstructured to Structured Information in Military Intelligence: Some Steps to Improve Information Fusion. *SCI Panel Symposium*, London.

[6] Bresnan, J. (2001). *Lexical-Functional Syntax*. Oxford, UK: Blackwell Publishers.

[7] Gruber, T. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5, 199-220.

[8] Hall, D. l. & Linas, J. (Eds.) (2001). *Handbook of Multisensor Data Fusion*. Boca Raton: CRC Press.

[9] Hecking, M. (2004). Information Extraction from Battlefield Reports. *Proceedings of the 8th International C2 Research and Technology Symposium*. Washington, DC: CCRP.

[10] Hoffmann, T. (1997). Ein Reduktionssystem zur Klassifikation von Feindbeobachtungsmeldungen. FFM-Bericht Nr. 481. Wachtberg: FGAN.

[11] Lorenz, F.P. (2003). Ein Ansatz zur rechnerbasierten Unterstützung bei der Analyse einer Hinterhaltsbedrohung aus Meldungsinformationen und Situationskontext, *FKIE-Bericht Nr. 62*, Wachtberg: FGAN.

[12] MIP website: http://www.mip-site.org.

[13] NATO (2003). NATO Interoperability Directive (AC322-SC/2-WG/4 WP(2003)0015-REV2).

[14] NATO Standardization Agency (NSA) (2003). AAP-6(2002) NATO Glossary of Terms and Definitions, http://www.nato.int/docu/stanag/aap006/aap6.htm.

[15] NATO (2002). AJP 2.0 Allied Joint Intelligence Counter Intelligence and Security Doctrine, NATO/PfP Unclassified, Ratification Draft 2.

[16] Schade, U. (2004). Automatic report processing. *Proceedings of the 9th International Command and Control Research and Technology Symposium*. Kopenhagen.

[17] Schade, U. & Frey, M. (2004). Beyond information extraction: The Role of Ontology in Military Report Processing. In: Buchberger, E. (Ed.), *KONVENS 2004: Beiträge zur 7. Konferenz zur Verarbeitung natürlicher Sprache (= Schriftenreihe der Österreichischen Gesellschaft für Artificial Intelligence, Band 5)* (pp. 177-180). Wien.

[18] Shieber, S.M. (1986). *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.

[19] Staab, S. & Studer, R. (2004). Preface. In: Staab, S. & Studer, R. (Eds.), *Handbook on Ontologies*. Berlin: Springer.

[20] Schwarzkopf, H.N. & Petre, P. (1992). *It Doesn't Take a Hero*. New York: Bantam.