한국지능정보시스템학회논문지 제10권 제3호 2004년 12월(pp.11~26)

An Agent based Emergency Warning System for Dealing With Defensive Information Warfare in Strategic Simulation Exercises

Yong-Han Lee
Department of Industrial and Systems Engineering,
Dongguk University
(yonghan@dgu.edu)

Soundar R.T. Kumara
Department of Industrial and Manufacturing Engineering,
The Pennsylvania State University, University Park, PA 16802
(skumara@psu.edu)

Software for analyzing documents on the net to detect specific categories of occurrences is in great demand. In the current world where detecting terrorist threats is critical there is a great need for such systems. One of the critical application areas of such software is the automatic detection of a national infrastructure emergency. In this research an agent-based generic architecture for emergency warning systems is proposed and implemented. This system, called the National Infrastructure Emergency Warning System (NIEWS), is designed to analyze given documents, to detect threats, and to report possible threats with the necessary information to the appropriate users autonomously. In addition, a systematic analysis framework to detect emergencies on the subject of defensive information warfare is designated and implemented through a knowledge base. The developed system along with the knowledge base is implemented and successfully deployed to Strategic Crisis Exercise (SCE) at the United State Army War College (USAWC), saving a good amount of money by replacing human SMEs (subject matter experts) in the SCE.

Key words : intelligent software agent, rule-based system, strategic simulation exercise, information warfare, emergency warning system

Received: July 2004 Accepted: December 2004 Corresponding Author: Yong-Han Lee

1. Introduction

Because of the ubiquitous computer and Internet access today, more and more information is available in the form of text on the net. Obviously some of this information may be extremely important, for example, evidences about possible threats to national security are one such information. However, analyzing the huge amount of documents on the net in real time is not only labor-intensive but also dependent on timing, i.e. detecting possible threats in an early stage is a critical requirement. Therefore, an intelligent system, which can autonomously analyze, extract, and report a given specific category of facts from the electronic documents, is in great demand in

different application areas. One of the critical application areas of such software is the automatic detection of a national infrastructure emergency.

We propose and develop an agent-based generic framework for an emergency detection called National Infrastructure Emergency Warning System (NIEWS). The proposed system architecture supports multiple agents so that multiple subject-specific agents (subject matter experts) can work simultaneously. NIEWS has a flexible architecture in the sense that users can generate different agents with different beliefs by assigning different expert knowledge bases. To verify the system in real situations, we generate a knowledge base on Defensive Information Warfare (IW-D) and show the propriety of the proposed system and the knowledge base.

The following section is devoted to the background research along with functional requirements for NIEWS. The proposed approaches for developing NIEWS are represented in Section 3. Sections 4 to 7 describe the details of some of important design and implementation issues, and Section 8 summarizes a real-life performance of NIEWS. Finally, we present conclusions in Section 9.

2. Background

The U.S. Army War College (USAWC) oversees the Strategic Crisis Exercise (SCE), a simulation exercise, every year. During this exercise, students take the role of different federal and non-federal organizations, and they are informed of events that are similar to real-world events. The objective of this exercise is that the students must learn to make correct decisions at critical times as they are required to do in the real world. In this exercise, students get advice from more than 120 *subject matter experts* (SMEs), bringing some measure of reality to the exercise. Most of the SMEs are, however, temporally hired and do not deliver advice unless contracted. For this reason, the USAWC would like to maintain the corporate knowledge of the SMEs for use during future exercises (USAWC 1998). On the basis of this necessity, we developed NIEWS, which can detect and report the possible threats to the students on behalf of human SMEs, and the IW-D was chosen as the first target subject to be implemented.

To develop the NIEWS framework, the functional requirements for the system were defined first. The essential requirements for NIEWS are:
to maintain expert knowledge including the subject-specific dictionary terms, known facts, and inference rules related to subject areas,

- to monitor and parse the electronic documents called the *master scenario event list* (MSEL), which are sent to the students according to the predefined scenario schedule,
- to analyze the newly parsed and extracted facts by using the expert knowledge and the current beliefs, updating these beliefs dynamically, and
- to send warning messages along with a brief analysis result to the appropriate recipients when a threat is detected.

3. Approaches

To satisfy the functional requirements mentioned above, we adopted the following five approaches from the perspectives of system architecture, knowledge representation, reasoning method, knowledge extraction, and knowledge maintenance, respectively.

Intelligent Software Agent Architecture : The basic system architecture of NIEWS is of an autonomous intelligent software agent, which senses the changes in the real world autonomously, reasons about a specific subject, and communicates with human users and/or agent peers. In addition, in order to meet the requirement for multiple subjects in the future, the system is designed as a multi-agent system, where the individual agents are in charge of their own subject matter. All the agents can work concurrently.

Attribute-based Knowledge Representation : To map the English sentences of original MSEL documents to the parameterized internal facts, we used attributed-based knowledge representation, where the contents of an MSEL are converted into a set of facts in parametric form using an editor offline

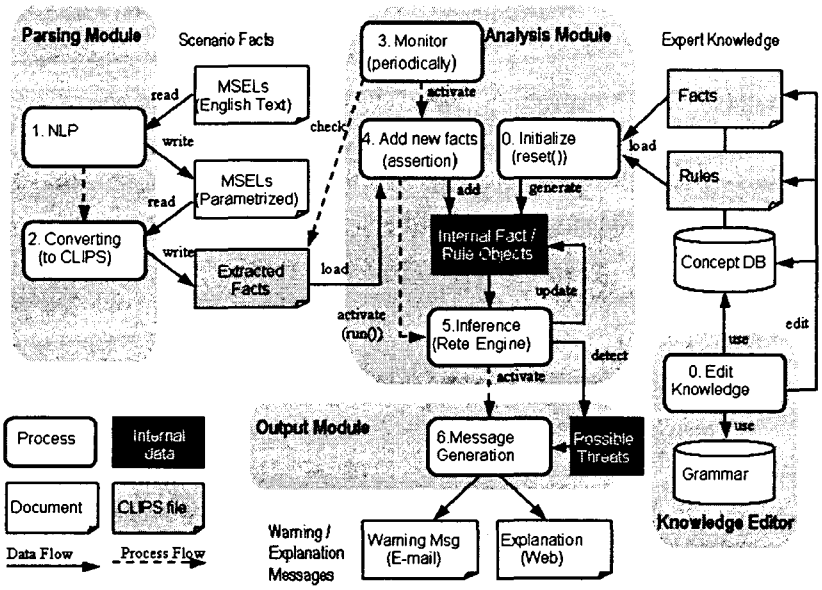
Rule-based System : The fundamental three categories of expert knowledge - dictionary terms, facts, and rules - must have representation schemes that are compatible with each other. The most adequate approach for this representation is a *production system*, which consists of factual knowledge and compatible conditional knowledge, i.e. if-then rules (Giarrantano and Riley, 1993). For the purpose of implementation, the expert knowledge was represented using CLIPS syntax, which can be processed by the Java Expert System Shell (Jess) API (Friedman-Hill, 1998).

Semantic Analysis and Knowledge Extraction : For more efficient management of the knowledge base, the domain expert knowledge was extracted and analyzed by constructing hierarchical relationships among concepts and classifying internal factual knowledge and production rules. These were done by investigating sample MSEL documents and by discussing with a domain expert. Finally, on the basis of the analysis results, we developed a systematic threat analysis framework.

Knowledge Maintenance using Knowledge Editor : The knowledge bases are managed through a software module called Knowledge Editor, which has the following functions: (1) editing the dictionary terms and their relationships, (2) editing the factual knowledge, (3) editing the production rules, and (4) maintaining integrity between dictionary terms, facts, and rules.

4. System Architecture

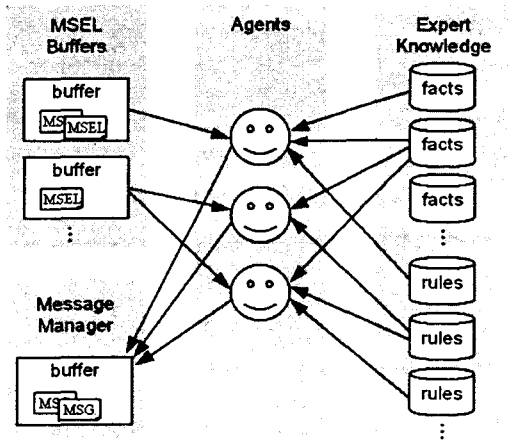
As mentioned previously, NIEWS adopted the rule-based production system model for knowledge representation and reasoning. [Fig. 1] shows the configuration of the NIEWS archi-



[Fig. 1] NIEWS system architecture - knowledge based system view

tecture along with its data and process flow from the rule-based production system's viewpoint. Knowledge Editor manages the expert knowledge base, and stores it as CLIPS-format files. When a new agent is created, the *initialization* process loads these files and converts them into internal facts and rules, thereby creating the initial belief set. At the same time the agent starts checking pre-defined input buffers, and reads in newly arrived MSELs to parse them into internal facts. These new internal facts are added to the list of existing internal facts. The inference engine automatically detects the changes in the internal facts list, and updates the list by applying the rules. If some rules that detect IW threats are fired, appropriate warning messages are generated based on the facts that cause them and sent to the recipients listed on a pre-defined mailing list. These processes keep running asynchro nously until a user destroys the agent.

[Fig. 2] shows the multi-agent perspective of NIEWS system architecture. When creating a new subject-specific agent, a user can assign: (1)



[Fig. 2] Multi-agent system view

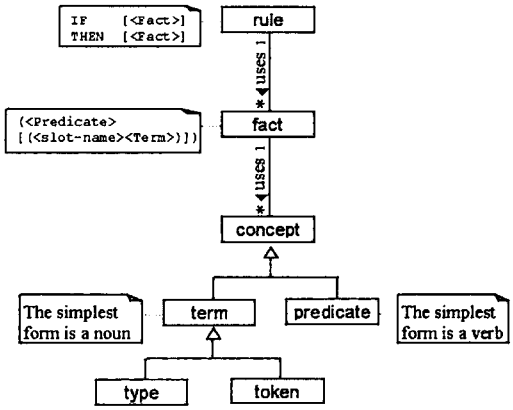
the path to the *input message buffer* where new MSEL messages arrive, (2) paths to the CLIPS files containing expert rules, and (3) paths to the CLIPS files containing expert's known facts. Because the agents are allowed to be linked to multiple CLIPS files, the expert rules and facts can be managed more systematically, and common knowledge among agents can be shared without duplication.

5. Knowledge Representation

As mentioned previously, we used parameterized facts represented in the form of an object-attribute-value structure. On the basis of the semantic analysis on sample MSELs and discussions with a domain expert, we generated the knowledge representation scheme for NIEWS.

5.1 Building Blocks - Dictionary, Facts and Rules

In general, the building blocks of a rule-based knowledge base are *dictionary*, *facts* and *rules*. The dictionary comprises a set of terms along with the information on their semantic and syntactic usage. In NIEWS, we classified the dictionary terms into *types* and *tokens*, which represent classes and instances respectively in the object-oriented perspective. Along with predefined *predicates* these dictionary terms are used to represent facts, which are used to represent rules as shown in [Fig. 3].



[Fig. 3] Knowledge Building Blocks

In NIEWS, every fact and rule is represented in CLIPS format. In CLIPS syntax, facts come in two categories, *ordered* and *unordered*. Ordered facts are just lists whose head must be a literal (neither variable nor expression) like (is_a korea country); Unordered facts are a combination of a head and following multiple *slots*, which are pairs of slot name and value, like (msel (pred fail)(who hospital)(how out_of_service)(by_what earth_quake)(where memphis) (confidentiality low)). Ordered facts are used for representing the relationship between dictionary terms because of their simple format, while the unordered facts are used for representing the real world events or status, including MSEL facts, because of their flexibility. Rules in CLIPS consist of a list of patterns and a list of actions, under the IF-THEN structure. The patterns are matched against the facts list. When facts are found that match all the patterns of a rule, the rule is ready to be fired, like other *forward chaining* expert systems (Friedman- Hill, 1998).

5.2 Predicates and Theirs Semantics

The ordered facts are used for representing the logical relationships among dictionary terms and consist of one predicate and two attributes - i.e. (predicate attribute-1 attribute-2). The NIEWS dictionary and knowledge base use three predicates for this type of facts including is_a, belongs_to and uses.

was defined based on the in-depth analysis of sample MSEL documents. This set of predicates can be extended using the Knowledge Editor if required. For each predicate, an appropriate set of attributes and their semantics was also defined.

5.3 Assigning Confidence Factor

Every fact that NIEWS learns from the real world has its information *source* and *source mode.* Every information source and source mode has its own confidence level. For example, CNN news and a high school news paper have obviously different levels of confidence as sources of information. Source mode or the way in which the information was given (for example, confirm or suspect) is another factor for differentiating the confidence level of information. Besides the information source and the source mode, every expert rule also has different confidence levels. Hence, for more precise analysis, a confidence factor (CF) representation scheme is required both for the facts and for the rules. The following is the CF representation scheme of NIEWS.

Initial CF Assignment on Facts: The initial CF of a fact can be determined based on source and source mode. We can assume that the CF of a source and a source mode are independent of each other. For example, the source mode 'confirm' has the same CF value irrespective of its coupled information source. This assumption lets us to calculate the CF of a fact by multiplying the CFs of the source and source mode, i.e., (CF of a fact) = (CF of the source) × (CF of the source mode). In order to assign a CF on a source or a source mode, they were clustered based on their levels of confidence. Each term, which can be used as a source or source mode, needs to be mapped into one of predefined clusters, which have their own CF values. This information can be represented by a simple CLIPS fact, such as (is_a nation_wide_network cf90_source), which means every source belonging to nation_wide_network has 90 percent confidence as an information source.

CF Assignment in Production Rules : Each production rule must have its own CF value because most of the rules cannot be proved one hundred percent true. The facts that match the patterns in a rule already have their own CF values, so the problem is how to calculate the CF value of the consequent fact. The basic idea of the calculation is that the CF of the consequent fact from conjunctive conditions must be dominated by the least reliable pre-condition, while the CF of the consequent fact from disjunctive conditions must be dominated by the most reliable pre-condition

(Nagao, 1990). A simple CF calculation method for the case of the two pre-conditions is as shown below:

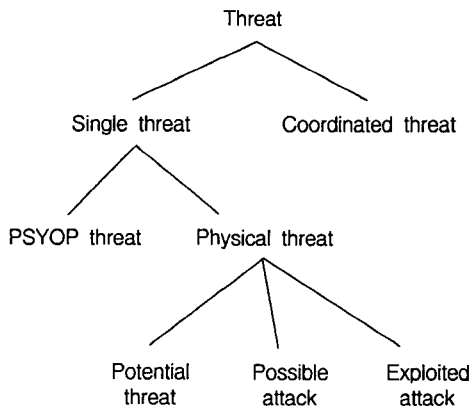
```
fact1 = true with CF = cf1
fact2 = true with CF = cf2
rule1: if fact1 ∨ fact2
    then fact3, with CF = cf3
rule2: if fact1 ∧ fact2
    then fact4, with CF = cf4
⇒
fact3 = true with CF = min(cf1,cf2) × cf3
fact4 = true with CF = max(cf1,cf2) × cf4
```

6. Expert Knowledge Analysis

Using the fact and rule representation scheme along with pre-defined predicates, we can represent the facts describing both the real world and the expert 'fact' knowledge. To represent the expert 'rule' knowledge, however, we need to analyze and extract the expert's knowledge of how to decide possible threats. In our application, the rules are thought of as mapping functions, which map the existing facts to new facts. For the systematic analysis and maintenance of the expert knowledge, the dictionary terms, facts and rules were classified. The following topics are discussed in this section: (1) classification of IW-D threats that we want to detect, (2) classification of facts and rules for systematic maintenance of expert knowledge, and (3) some details of *decision rules*, which finally decide possible threats. The knowledge explained in the following subsections were extracted from human experts and corresponding literature including (Albert, 1996), (Anderson, 1998), (Libiki, 1996, 1997), and (March, 1997).

6.1 IW-D Threats

IW threats are classified as shown in [Fig. 4]. An IW threat can be either a single threat that intends a short-term effect by itself or a part of interrelated and more strategic IW attacks. In both cases, a single instance of IW threats that have been made in a specific time and place are called *single threats*. Meanwhile, a set of interrelated IW threats having a single purpose is called a *coordinated threat*. In this paper, single threat analysis framework is mainly discussed.



[Fig. 4] IW-D threat classification

The single threats again fall into two subcategories, *physical threats* and *psychological operation* (PSYOP). Physical attacks aim at damag-

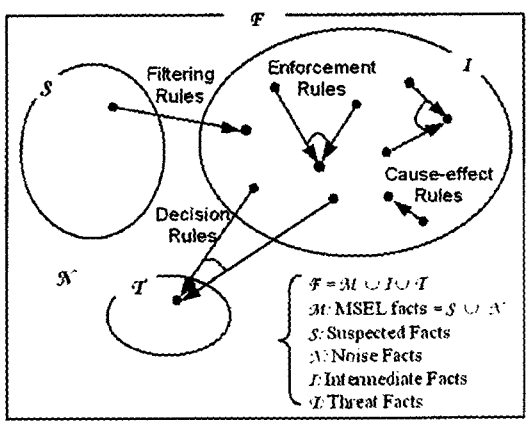
ing a physical national infrastructure using IW weaponry or means; PSYOPs aim at people's opinions, for example, causing people to distrust their government's capability to protect them. Physical threats can be sub-categorized into three threat classes according to the level of significance of the threat (NIEWS deals with more detailed classification of threats): *potential IW threat*, *possible attack*, and *exploited attack*.

6.2 Classification of Facts and Rules

To maintain a huge and growing set of facts and rules, a systematic view of them is required. Thus, we classified facts and rules for this purpose. The advantages of the classification is that we can maintain the facts and rules more systematically, can reduce the number of facts and rules, and can simplify the rules by reducing the length of patterns in the rules.

At first, we classified facts into three main categories: *MSEL facts*, *intermediate facts* and *threat facts*. MSEL facts are basically parameterized-MSEL statements themselves. Among them, only some facts are related to IW. If some MSEL facts are found to be IW-related, they are converted into intermediate facts and used for generating any meaningful conclusions. The threat facts describe the possible or exploited IW threat inferred from intermediate facts. [Fig. 5] shows the different sets of facts along with rules as mapping functions. We also classified the rules into four main categories: *filtering rules*, *cause-effect rules*, *enforcement rules*, and *decision rules*. The rules can be thought of as mapping functions defined in the space of facts as shown in [Fig. 5]. The roles of the rules are described as follows.

- **Filtering rules** are used to filter the MSEL facts to generate intermediate facts, which hold meaningful information regarding IW-D. These filtering rules simply check if an MSEL fact contains any suspected concept.
- **Cause-effect rules** are used to derive another intermediate fact from one or multiple intermediate facts based on common sense and/or expert knowledge.
- **Enforcement rules** control the CF values of intermediate facts when redundant and/or repetitive facts are found for a single event. These rules are also helpful to keep the agents' working memory compact by reducing redundant information.
- **Decision rules** generate threat facts from intermediate facts by checking the possibility of an actual threat in the following points of views:



[Fig. 5] Classification of facts

motivation and capability of enablers, importance and vulnerability of targets, and level of damages.

6.3 Decision Rules for Single Threats

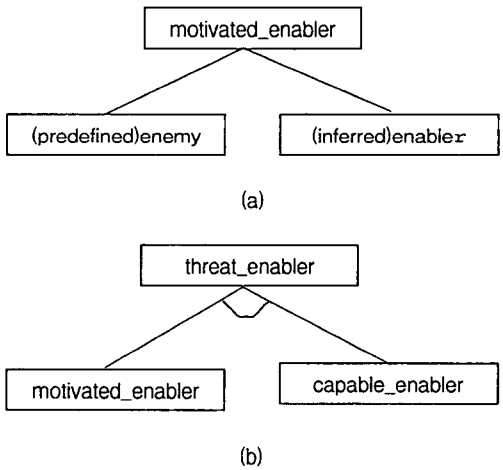
The decision rules for a single threat take into account the following six criteria - *motivation* of enablers, *capability* of enablers, importance of *targets*, *vulnerability* of targets, level of *damages*, and level of *confidentiality* of the information. The decision rules have different combinations of the levels of these factors as preconditions to map each threat into a specific category of threats. The levels of each factor are shown in <Table 1>.

<Table 1> Criteria of single threat decision

Factors	Criteria	Levels for Decision
	Motivation	{ motivated_enabler ¬ motivated_enabler
	Capability	{ capable_enabler ¬ capable_enabler
Target	Importance	{ national_infra ¬ national_infra
	Vulnerability	{ vulnerable ¬ vulnerable
	Damage	{ serious ¬ serious
Information	Confidentiality	{ High: *classified/intelligence* Mid: *non-classified/private* Low: *publicly known*

The enablers of threats are categorized into three groups - *motivated enabler*, *capable enabler* and *threat enabler*. As shown in [Fig. 6] (a), motivated enablers comprise either predefined enemies or inferred motivated enablers, who are identified during the analysis. For example, the latter kind of enabler can be identified if an MSEL says that the enabler (a country, a group or a person) warned a target of concern with IW attacks. The capable enablers are the ones who have IW weaponry and/or development capability. If a country or a group is found to be both motivated and capable, it is called a threat enabler as shown in [Fig. 6] (b).

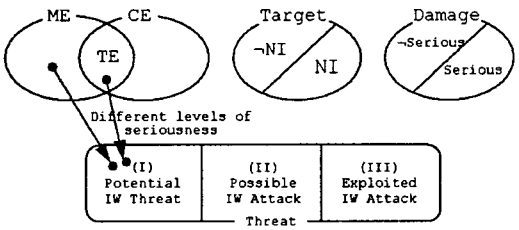
The following are some of the details of the decision rules to detect IW-D threats based on the above criteria.



[Fig. 6] Logical relationships of enabler concepts

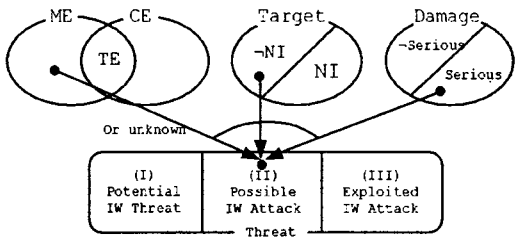
- **Decision Rules for Potential Threats :** This decision can be made by detection of simple threats or warnings aimed at the targets of concern. Hence, the only necessary pre-condition is that the actor should be an *inferred motivation_enabler*. However, in using additional information on the enabler's IW capability, two different levels in the seriousness of the threat

can be determined as shown in [Fig. 7]. In the figure, ME, CE, TE and NI denote motivated enabler, capable enabler, threat enabler, and national infrastructure, respectively.

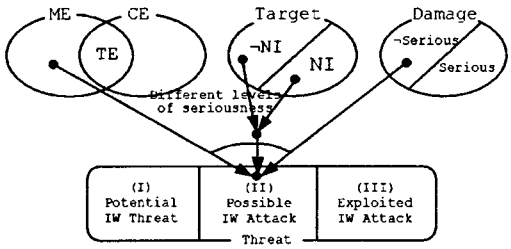


[Fig. 7] Potential threat decision

- **Decision Rules for Possible IW Attacks** : This decision can be made by detection of physical damage that is believed to be a result of IW attacks. Depending on the category of the target and the seriousness of the damage, different levels of possible IW attacks can be determined as shown in [Fig. 8]. The possible combinations are: (1) serious damage to non-national-infrastructure, (2) non-serious damage to non-infrastructure, and (3) non-serious damage to national infrastructure.
- **Decision Rules for Actual Physical IW Attacks** : This decision can be made by detection of serious physical damage caused by IW attacks on national infrastructure, as shown in [Fig. 9]. A higher confidence value is assigned if the enabler can be clearly identified as a threat_ enabler.

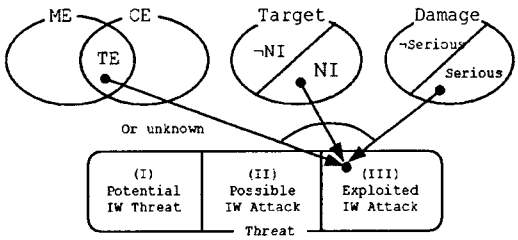


(a) Level- I possible IW attack



(b) Level-II possible IW attack

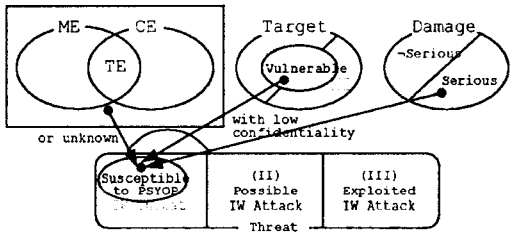
[Fig. 8] Possible IW Attack Decision



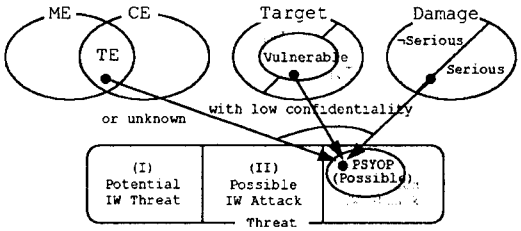
[Fig. 9] Actual physical IW attack decision

- **Decision Rules for PSYOP IW Threats** : In deciding PSYOP threats, confidentiality of the information and vulnerability of the target must be taken into account. [Fig. 10] shows three typical cases of a PSYOP decision: (1) any national infrastructure gets seriously damaged by a natural disaster. In this case, the public might

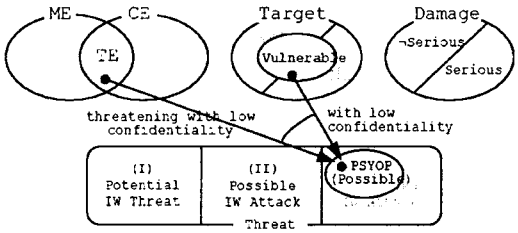
believe that their government does not have the capability to prevent or to recover from the disaster efficiently. Then we can conclude that the public is *susceptible* to a PSYOP from motivated enablers, (2) an actual IW attack was exploited, and it was also known publicly that the target was vulnerable to that kind of IW attack. Then it can be interpreted that the enabler intended the PSYOP effect as well as the physical attack (or intended only the PSYOP effect), (3) a threat enabler threatened the government with very low confidentiality (for example, through mass media), and the target is believed to be vulnerable to that kind of IW attacks. This is the most direct PSYOP to the public no matter whether actual IW attacks follow or not.



(a) Susceptible to PSYOP



(b) Possible PSYOP

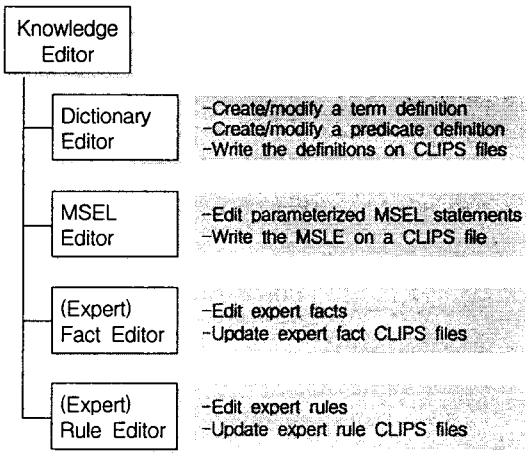


(c) Highly possible PSYOP

[Fig. 10] PSYOP IW threat decisions

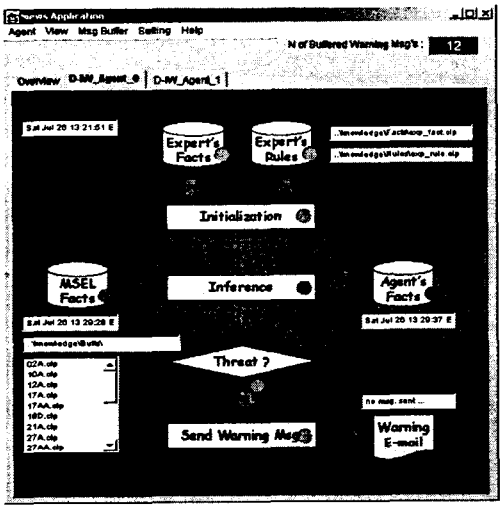
7. Knowledge Editor

The Knowledge Editor is an offline program that maintains the expert knowledge, including dictionary terms, expert facts and expert rules, so it can be considered as a system of three distinctive sub-modules - *dictionary editor*, *fact editor*, and *rule editor*. Also another offline knowledge editor, called *MSEL editor*, was developed to edit MSEL messages in CLIPS format. The essential functions of these modules are shown in [Fig. 11].

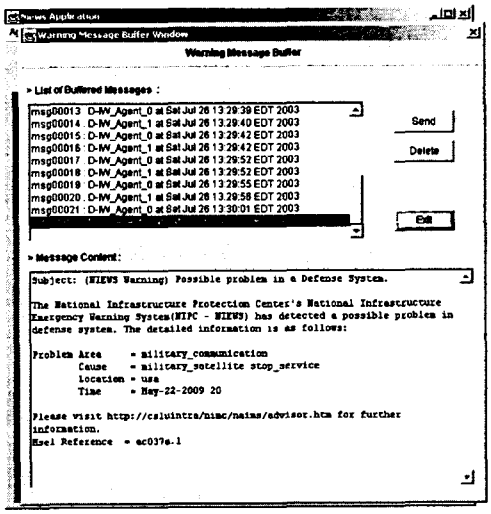


[Fig. 11] Modules and functions of knowledge editor

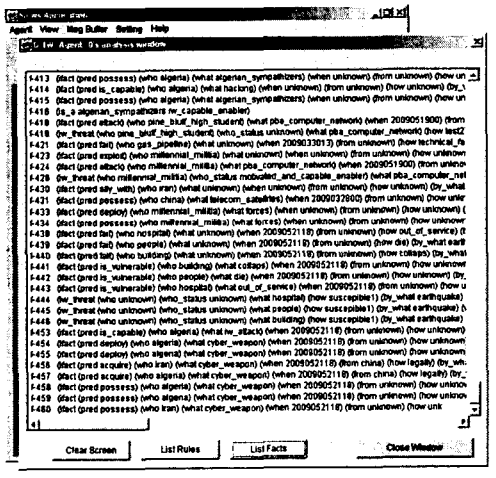
The NIEWS knowledge base is based on "closed world assumption" for the purpose of referential integrity. Namely, every fact and rule must be defined by the terms and predicates that are already defined in the knowledge base. In order to maintain the data integrity based on this assumption, the editors other than the dictionary editor never allow users to key in new terms and predicates. Instead, users can select one from the list of terms, which the editor proposes based on the semantic definitions of the input fields. When a new term is required to describe a fact, dic-



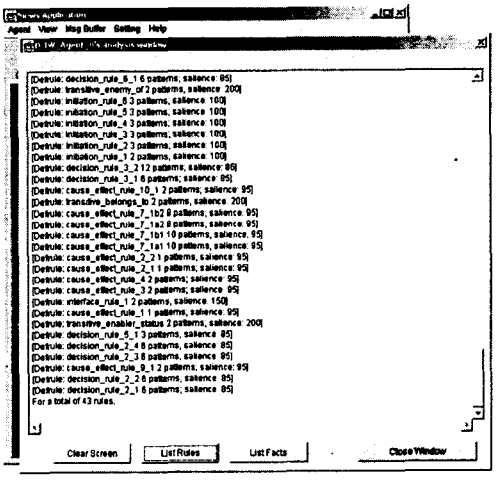
(a) Status of an agent



(b) Warning message-buffer window



(c) List of facts



(d) List of rules

[Fig. 12] Snapshots of the NIEWS control panel

tionary editor is automatically spawned for users to create the new term. The dictionary data structure, which is a set of interrelated trees and linked lists, is dynamically updated even in this case, maintaining data integrity.

8. Implementation and Deployment

NIEWS is implemented using Java and Jess API. Jess is an expert system shell and a set of API, which allows other applications to embed this shell. Because Jess was written in Java, it can be easily integrated with Java applications like NIEWS. To allow multiple agents to work simultaneously, we implemented the individual agents as separate threads.(Bigus and Bigus, 1997) [Fig. 12] shows the snapshots of NIEWS main control panel. As shown in [Fig. 12] (a) each subject-specific agent's status can be monitored on each tab panel in the system. Generated warning messages are stored temporarily, waiting for the final decision of NIEWS administrator before sending to appropriate persons as shown in [Fig. 12] (b). Dynamically-updated beliefs (including rules and facts) of the agents can be browsed using pop-up windows as shown in [Fig. 12] (c) and (d).

The current version of NIEWS and the IW-D knowledge base were deployed in the SCE. The content of the knowledge base are summarized in <Table 2>.

<Table 2> Content of the Knowledge Base in the SCE

Category		Size
Expert Facts		283
Expert Rules	Filtering Rules	6
	Cause-Effect Rules	18
	Decision Rules	12

During the exercise, four IW-D scenarios were deployed among many other categories of scenarios. The scenarios consist of multiple MSELs, and each MSEL contains multiple statements. <Table 3> shows the sizes of the scenarios.

<Table 3> Scenarios deployed in the SCE

Scenario	Number of MSELs	Number of facts
A	3	11
B	4	8
C	12	31
D	10	25

NIEWS successfully detected the new MSEL messages, analyzed the facts of the MSELs, and generated adequate warning messages, totaling 19 messages, without any collision for the randomly arriving MSELs. A typical example of the warning messages generated by NIEWS is shown in [Fig. 13].

The responses from the students who were advised by the NIEWS with these warning messages were very positive. Some weaknesses, however, were pointed out too. The two major issues from them are (1) lack of an explanation subsystem, which explains the details of the threat including backdrops and related reference in-

Subject: (NIEWS Warning) Highly Possible IW Attack in Progress by millennial_militia

The National Infrastructure Protection Center's National Infrastructure Emergency Warning System (NIPC - NIEWS) has detected a possible IW attack is preparing/progressing. The detailed information is as follows:

Threat Type = single threat as analyzed so far
Actor = millennial_militia (who has both motivation and IW-capability)
Target = pba_computer_network
Action= attacked or tested its capability, but the damage is not severe. (weapon = illegal_penetration)
Location = little_rock
Time = May-19-2009-00

Please visit http://csluintra/nimc/naims/advisor.htm for further information.
Msel Reference = ms017a.1 ms012a.1

[Figure 13] An example of the warning messages

formation, and (2) no feedback mechanism, by which the students' reaction to the threats can update the agents' beliefs.

9. Conclusions

In this research, we developed a complete solution to support the students on behalf of human SMEs in SCE at USAWC as follows. Although the adopted technical approaches are not unique to this research, they are well organized in the proposed system architecture, and fulfill the SCE requirements successfully.

- **Autonomous Agent Architecture:** We developed an agent based generic emergency warning system architecture called NIEWS. The autonomy of the agents makes this system work truly on behalf of human SMEs.
- **Multi-agent / Multiple knowledge bases:** NIEWS architecture allows agents to be linked to multiple CLIPS files so that these knowledge base files can be managed in a flexible and robust way.
- **CLIPS based Knowledge Representation:** We adopted the popular language, CLIPS, to represent knowledge in NIEWS, taking full advantage of the rule-based production system.
- **Systematic Knowledge Maintenance Mechanism:** The approach for maintaining compact knowledge bases is based on our novel classification of facts and rules. The proposed knowledge management framework can be easily applicable to other subject areas.
- **IW Emergency Analysis Framework:** We proposed an emergency analysis framework in the subject of IW-D, and implemented it in the IW-D knowledge base. We clearly identified the criteria and the decision rules for deciding IW threats.

NIEWS was developed for an instructional purpose for USAWC, but its system architecture and the emergency analysis framework can be used in the real world emergency warning system.

In order for NIEWS to be used in real world situations in the future, the NLP (Natural Language Processing) module should be one essential part of the research, even though this module was excluded from the current version. By adding the NLP module, which replaces the MSEL editor, the system will be a truly autonomous system. In NLP module development, our work on semantic definitions of predicates and corresponding attributes will play an important role.

Acknowledgement

본 연구는 미육군 전쟁대학(The US Army War College)의 지원을 받아 수행되었음.

References

[1] Albert, D. S., *Defensive Information Warfare*, Institute for National Strategic Studies, National Defense University, Washington DC, 1996

[2] Anderson, K., "Intelligence-based Threat Assessments for Information Networks and Infrastructures", URL: http://www.aracnet.com/~kea/Papers/threat_white_paper.shtml, 1998

[3] Bigus J. P. and Bigus J., *Constructing Intelligent Agent with Java* (2nd Ed.), Wiley, 2001

[4] Friedman-Hill E. J., *The Java Expert System Shell* (verson 5.2), SAND98-8206, Sandia National Laboratories, 2001

[5] Giarrantano, J. and Riley G., *Expert Systems - Principles and Programming* (3rd Ed.), Brooks Cole, 1998

[6] Libicki, M. C., *Defending Cyberspace*, Institute for National Strategic Studies, National Defense University, Washington DC, 1997

[7] Libicki, M. C., *What is Information Warfare?*, Institute for National Strategic Studies, National Defense University, Washington DC, 1996

[8] March, R. T., *Critical Foundations-Protecting America's Infrastructures*, President's Commission on Critical Infrastructure Protection, October 1997

[9] Nagao, M., *Knowledge and Inference*, Academic Press, 1990

[10] USAWC, "The Strategic Crisis Exercise" (a presentation material), Center for Strategic Leadership in U.S. Army War College, June 1998

요약

전략시뮬레이션 훈련에서의 방어적 정보전을 위한 에이전트 기반 위기경보시스템의 개발

이용한* · Soundar R.T.Kumara**

특정한 사건의 발생을 감지하기 위해서 인터넷 상의 문서들을 분석하는 소프트웨어는 매우 유용하다. 특히, 오늘 날과 같이 테러의 위협을 감지하는 것이 결정적으로 중요한 시점에서 이러한 시스템의 필요성은 더욱 크다고 할 수 있다. 이러한 소프트웨어의 중요한 응용분야 가운데 하나는, 자동으로 국가 기반의 위기를 탐지해 주는 시스템일 것이다. 본 논문에서는 위기 경보 시스템을 위한 에이전트 기반의 일반적 아키텍처를 제안하고 구현하였다. NIEWS (National Infrastructure Emergency Warning System)라고 명명된 이 시스템은 주어진 문서를 분석하여, 위협을 감지하고, 가능성 있는 위협들을 필요한 정보와 함께 적절한 사용자에게 자동으로 보고해 주도록 설계되었다. 이와 아울러, 본 연구에서는 방어적 정보전에 관련된 위기를 감지하기 위한 체계적인 분석 틀이 고안되고 지식베이스로 구현되었다. 본 연구에서 개발된 시스템과 지식기반은 실제로 구현되고, 미육군 전쟁대학의 전략 시뮬레이션 훈련인 SCE (Strategic Crisis Exercise)에 설치되어 활용 되었으며, 여기서 SCE에 필요한 고비용의 분야별 전문가 (SME: subject matter expert)들을 대체 함으로써 많은 비용을 절감할 수 있었다.

* 동국대학교 산업시스템공학과
** Department of Industrial and Manufacturing Engineering The Pennsylvania State University