computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific Journal of Computing

A SCALABLE INFERENCING SYSTEM FOR CIVILIAN TERRORISM INTELLIGENCE

Thomas J. Wheeler [1], Karpagavalli Vel [2]

1) University of Maine, Orono ME(USA), wheeler@umcs.maine.edu, http://www.cs.umaine.edu/~wheeler/
2) University of Maine, Orono ME(USA)

Abstract: *This paper describes an approach to developing a scalable intelligence inferencing system for civilian terrorism intelligence. There is an obvious need for such a system in light of failures in the intelligence community leading to the September 11th attacks. It is intended as a supplement to human intelligence analysis; while intelligence analysts are good at what they do, it's hard to see what information is important and integrate it. Information and inferences don't always flow up the chain of command and don't always get to where they are needed. Automated assistance can aid intelligence analysts, and managers helping to prevent other tragedies from occurring.*

The work explored an approach to automate (or provide assistance for) information fusion which effectively makes inferences over huge amounts of information and number of events using a scalable architecture. It is based on number of technical thought patterns: (1)evolutionary development of a system; (2)the use of layered inference graphs and tree based interpretations of them; (3) combining top-down with bottom-up inference; (4) and pattern matching with (un)certainty and importance calculations; (5)explanation based user interaction; and (6)using spatio-temporal localization, extrapolation/simulation and parallelism to raise inferencing performance to acceptable levels.

Keywords: - *Scalable Inferencing*

1. INTRODUCTION

Organizations need automated assistance in developing, and working with, a high level abstract view of existing and pending crisis situation. Its view of this situation must be attuned to the dual needs of planning and controlling the organization's operation in that situation. These systems need to fit with the organization's manner of operation and synergies with their visualizations, conceptualizations and interactional skills. The system must provide visualization and conceptualization assistance at multiple levels, the object, situation, and operation levels. The system must afford controllers with interaction capabilities with the displayed information for exploratory and intent developing reasons, and simulation (extrapolation) of future scenarios.

The creation, management, and display of these types of perceptual and conceptual assistance require significant information display processing, geospatial & semantic reasoning processing, database abstraction creation and database performance. Distributed processing and high performance computing nodes are integral to the provision of these capabilities. Cost effective provision of these capabilities depends on the use of COTS components and standardization. An approach to combining cost effectiveness with performance is to develop an infrastructure to be shared among the organization's systems, pushing the geospatial & semantic reasoning processing, abstract database functionality and database performance structures into that infrastructure. The infrastructure can be built out of COTS hardware and software supplemented with intelligent, performance enhancing software within the distributed processing architecture.

The current way of doing business is that commanders receive information manually fused by analysts and staff and transmitted, by official messages, through a hierarchical command and control structure. The information consists of "raw" data, analyses describing inferences, validation information,... This is combined with geographic and personal/hostile organizational information within that structure, and plans are then made based on the situation information and the organization's experience.

This paper describes an approach to developing a scalable intelligence inferencing system for civilian terrorism intelligence. There is an obvious need for such a system in light of failures in the intelligence community leading to the September 11th attacks. It is intended as a supplement to

human intelligence analysis; while intelligence analysts are good at what they do, it's hard to see what information is important and integrate it. Information and inferences don't always flow up the chain of command and don't always get to where they are needed. Automated assistance can aid intelligence analysts, and managers helping to prevent other tragedies from occurring

1.1 BACKGROUND

Following the September 11th events it has become clear in the United States that identifying potential terrorists before they execute terrorist acts is a major concern if future attacks are to be prevented. While intelligence analysts are good at what they do; but it's hard to see what information is important, difficult to make connections within the flow of information up the chain of command, and difficult to make sure that information and inferences get to where they can be made use of. We are focussing on a small number of issues surfaced in analyses that have been conducted of the tragic Sep.11 event: huge amount of data/events, insufficient analysis personnel, chain of command organization of intelligence processing, and unreliability of person to person inference communications.

There is a need for a system which would provide the inference infrastructure so that appropriate places in the organization can detect the possibility of terrorist threat. Our goal is a distributed, compartmentalized inferencing environment for civilian terrorism intelligence; providing the integrated inference and communications infrastructure to supplement to human intelligence analysis. In accordance with the evolutionary approach which we use, we have built the first version of the system as a prototype, using an expert system technology.

1.2 TECHNICAL BACKGROUND

The research has been guided by a number of technical thought patterns. Among these are (1)evolutionary development of a system; (2)the use of layered inference graphs and tree based interpretations of them; (3) combining top-down with bottom-up inference and pattern matching (4)with (un)certainty and importance calculations; (5)explanation based user interaction; and (6)using spatio-temporal localization, functional compartmentation, extrapolation/simulation and parallelism to provide scalability and raise inferencing performance to acceptable levels.

Evolutionary development is useful when the eventual functionality of a system is not known at the beginning of the project. Usage of the system is essential in order to develop the necessary understanding of the system's functionality and its use. The structure of fusion systems coalesces and abstract "raw" data into meaningful, more abstract objects, events, courses of events, strategies, etc. It organizing representations of these entities into layers, with the raw entities in the bottom layers and progressively more abstract entities in the upper layers.

The main ideas in achieving validity (and avoiding suspicion of politicizing) in inferencing, is to develop rules using validity fostering meta-rules like "past predicts future" and cause-effect, to use an open source strategy to rule development with "red teaming" tests, and to overlay nonfunctional attribute (like importance and trustworthiness) calculation over the inferencing framework, calculating and evaluating these attributes of the higher level nodes as properties or attributes of lower level nodes.

This paper describes our approach to achieving scalability, performance and validity in intelligence inferencing(fusion) by:

(1)Inferrencing information flowing upward, while creating the inferencing graph upward, from the bottom toward the top (forward chaining); or downward, from the top toward the bottom (backward chaining), or a combination of both.

(2)The inferencing takes place on a distributed network of computer clusters, compartmentalized by location, type of information and/or type of threat.

(3)Creating a unified spatial/semantic/temporal database (model) infrastructure.

(4)Migrating low level fusion/filtering down to sensors and intelligence network.

(5)Fusing at progressively higher levels in layers of abstraction; inferring progressively, (compound and abstract) object(s), situation(s), organization, intent and threat(s), from events and information.

(6)Incorporate validity and trustworthiness into inferencing in a way which permits optimistic and pessimistic inferences, as the situation warrants.

(7)Create a user interfacing strategy with perceptual enhancements to focus users on characteristics which interpret the situation and its unfolding effectively.

2. APPROACH

In accordance with the evolutionary approach, we have built the first version of the system as a prototype, using an expert system languages, Prolog. Prolog is a goal-oriented logic programming language using pattern matching(unification), tree-based data structuring

T. J. Wheeler, K. Vel / Computing, 2003; Vol. 2; Issue 2; 141-150

and automatic back tracking. It encourages programmers to describe situations and problems, not the means by which the problems are to be solved. In this way the programmer is more concerned with the knowledge and less concerned with algorithms that exploit the knowledge.

This prototype uses expert/knowledge-based system techniques. We envision a system that behaves like a distributed collection of experts in the intelligence analysis field, that fully understand their communications and ramifications of it. The system should provide the following functions: event and information acquisition and analysis; an appropriate database infrastructure with a semantic based abstract interface; an abstract layered (event&information -> object -> situation -> threat) fusion system; a problem solving function capable of using domain-specific knowledge and uncertainty; and an intelligence analysis, threat alert, visualization, user interaction system, which includes explanation of the system's intentions and decisions during and after the problem solving process.

Some recent developments afforded an opportunity for this system. First, fusion techniques have been developed to extract more abstract and more valid information from multi-source data by making inferences and using constrained reasoning. Second, displaying techniques have been developed which foster perceptual and conceptual recognition, envisioning and planning. Third, abstraction mechanisms are able to provide more valid compositions of higher level objects, recognition and predictions of paths and intentions of these objects and other multilevel organizational concepts. Fourth, database technology has developed spatial, semantic, and object-oriented techniques and systems which can provide cost effective underpinnings for these types of systems.

We use layered inference graphs and tree based interpretations of them. We combining top-down with bottom-up inference and pattern matching with (un)certainty and importance calculations. Explanation based interaction provides trustable intelligence analysis, threat alert and visualization. Spatio-temporal localization, compartmentation, extrapolation/simulation and parallelism provides scalability and raises inferencing performance to acceptable levels.

3. DESIGN DECISIONS

(1) Evolutionary development is useful when the eventual functionality of a system is not known at the beginning of the project and usage of the system is essential in order to develop the understanding of the system's functionality and its use necessary for the creation of the system. In using evolutionary development, one develops a system in a sequence of small increments, starting with a simple version of the envisioned system and adding functionality in small increments while using the system in (as close to possible to) an operational manner. The increments of functionality are guided by the (successes and failures in) usage of the system, and the structure of the evolving system is guided by anticipating and accommodate changes that are expected to occur during this evolution. Here, the main changes are expected to be patterns of information and inference at a number of levels, which are going to have to be discovered and validated by working with intelligence analysts, engineers and higher level users of the system. Other general classes of likely candidates for changes are the human interface, the inference techniques, and performance engineering, and thus, the system should be designed to accommodate changes in these areas.

(2) The approach to structuring fusion systems outlined here is to coalesce and abstract "raw" data into meaningful, more abstract objects, events, courses of events, strategies, ..., based on organizing representations of these entities into layers, with the raw entities in the bottom layers and progressively more abstract entities in the upper layers. Overlaid on this layering scheme, the entities are organized into a graph, usually a directed acyclic graph ("DAG"), and higher level, abstract nodes are created from lower level less abstract nodes by using patterns describing relationships.

Here we organize events, courses of events, objects, and abstractions of these into a graph of nodes (fig. 1) over which inference algorithms will run, producing and validating abstract nodes.

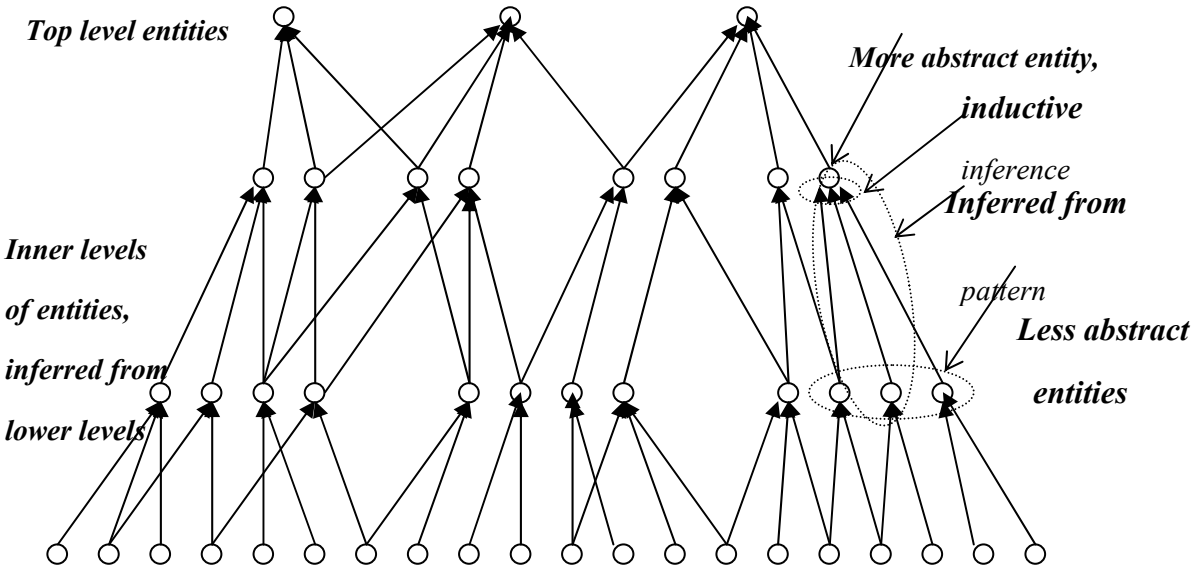
The inferencing algorithms integrate geographic interpretations with semantic interpretations of data, interpreting the graph from both perspectives. An example of a situation (fig 2) and how it is mapped onto this inferencing technique(fig 3) is included to provide a grounding of the rather abstract inference graph in fig 1. A prototype of this approach, written in PROLOG and run using a freeware interpreter, is included to validate the approach and show the encoding of the graph as patterns, or in AI terms, rules.

In the example[Time], a group of individuals arrives together and buys first class tickets. This raises a low level alert locally(Logan terminal). The first response is to trigger plain clothes security to wander into the area and assess the situation. Meanwhile, the next level nodes try to make inferences pertinent to the Logan situation, and when a certain level of probability inference of

hijacking is triggered, a medium level alert is displayed and sent up and down the communications network. When the top level combines that with the known scenario of crashing a jumbo jet into a building, a high level alert is triggered at the top level and at the Logan terminal. The flight is postponed and the suspected persons are detained.

Notice that the action is initiated as soon as possible, thus it can unfold in a controlled way. It interferes as little as possible with commerce flow. Defense is in progressive layers. If the alert was not valid, the response causes minimal inconvenience. Also notice the exact scenario does not need to be predicted, just the class of scenarios which lead to the same response.

Struture of Inferencing (Heirarchical Graph)



First level entities: observations [events, things].

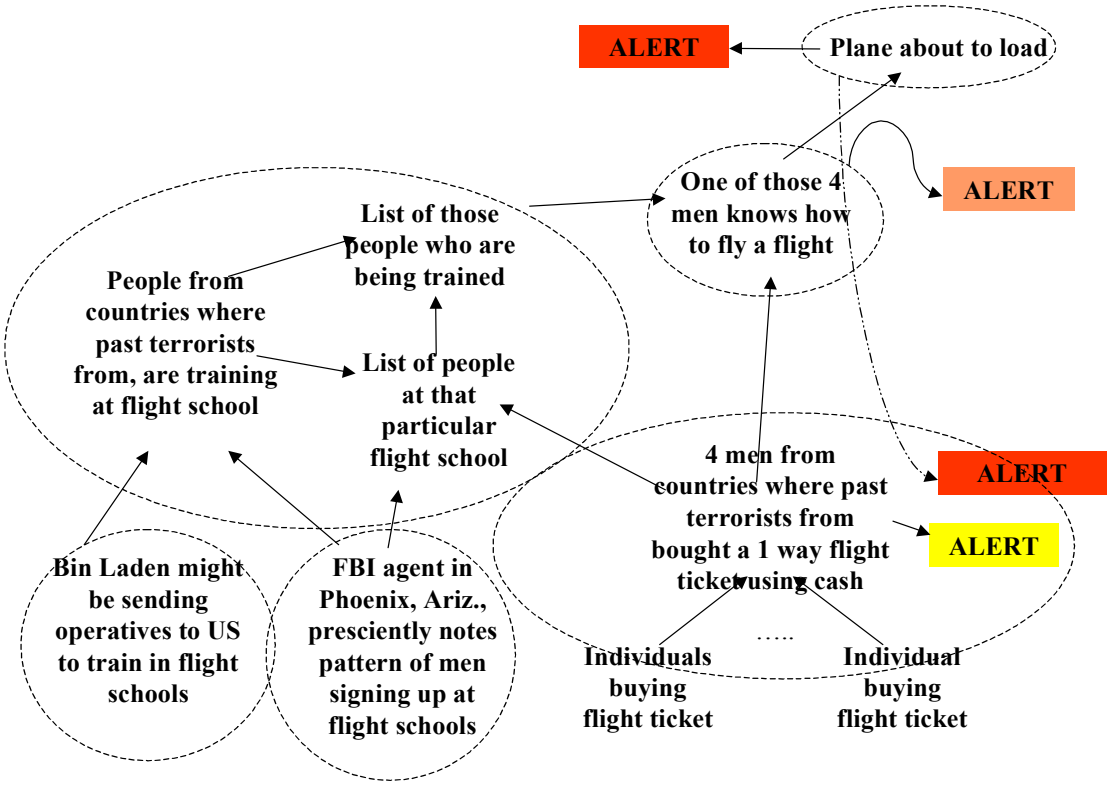


Fig.2

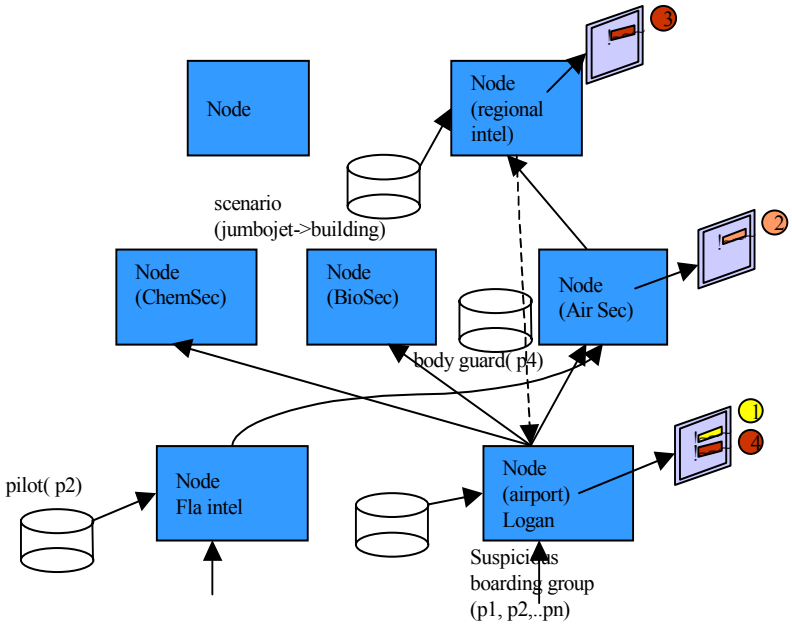


Fig.3

(3) It is important in understanding this approach to notice that while, in all of the figures, the inferencing information (the arrows in figs 1,2,3) flows upward, the creation of the graph can occur either upward, from the bottom toward the top, downward, from the top toward the bottom, or a combination of both. In bottom-up processing, the algorithm aggregates or abstracts lower level elements into higher level elements, going in the same direction as the inferencing information (in AI this is known as "forward chaining", forward reasoning or deduction), with each upper entity being a function of some lower level entities. In top-down processing, the algorithm infers a higher level entity, called "goals", from a pattern of lower level entities, going in the opposite direction as the inferencing information (in AI known as "backward chaining", backward reasoning or induction), and it does this by matching a pattern of lower level entities.

We combine bottom-up and top-down inferencing algorithms on a directed accyclic graph which grows at the bottom (the raw data) over time as new data arrives. The graph will be looked at as a forest of trees, and each tree type identified with a pattern (inf -> data1, data2, ..) where the left hand side (inf) is identified with the root of that tree and the right hand side (data1, data2, ..) is identified with its leaves.

The bottom up algorithms construct trees by taking a sequence of lower level entities that matches the leaves of a known inference tree, or equivalently the right hand side of a pattern (e.g. (data1, data2, ..)) and asserting the existence of an entity of that inference tree's root type, or equivalently the left hand side of a pattern (e.g. inf). The basis of its implementation is a list of first members of right hand sides which it compares to existing nodes to begin the matching process.

The top down algorithms look for patterns (e.g. upper -> lower1, lower2,..) by positing the existence of upper level entities (e.g. upper) and attempting to discover evidence for their existence by finding a pattern of lower level entities (e.g. lower1, lower2,..) in the known entity structure. The basis of its implementation is a list of left hand sides which it compares to existing nodes to start its matching process. (It should be noted that both strategies suffer from "explosion" of computation, bottom up generates all inferences, even ones we don't care about; top down looks for inferences it can't find. These will be addresses in the section on constraining search below.)

The technologies that are candidates for implementing this technique are: top-down inferencing techniques from AI e.g. techniques used by the PROLOG community, bottom-up techniques used by the logic database communities e.g. DATALOG and those used by the LISP community, language processing techniques i.e. Syntax directed compiling and attribute grammar technologies from the compiling communities.

(4) In creating the more abstract nodes there are other aspects of the information (usually non Boolean), beyond that used in the patterns (basically a Boolean technique), which must be

addressed in order to make useful and valid inferences. The data and inferences from it vary in importance, and a useful system will find important inferences before those which are less important. Likewise, some data and inferences are more trustworthy than others and a useful system will take this into account also. The approach here is to overlay the importance and trustworthiness calculation over the inferencing framework, calculating and evaluating importance and trustworthiness of the higher level nodes as properties or attributes of the nodes. As the nodes are discovered, by the (top-down or bottom-up) inference algorithms, importance and trustworthiness attributes are calculated as a function of the importance and trustworthiness attributes of higher or lower level nodes. The importance and trustworthiness calculations may be calculated, and often are, bottom-up, whether the inferences are done top-down or bottom-up. Other attributes such as contextual attributes, like inferring equipment from unit type may be just the opposite, normally being calculated top-down, even when the existence of the unit and its type is inferred bottom-up. (Techniques used in programming language compilers to evaluate attributes will probably be used here.)

(5) Human users of automated fusion must be able to effectively and efficiently make use of the fused information and, thus must be able to understand and trust the inferences made by the fusion software. The use of graph and pattern based inference techniques provides the basis for natural explanations of inferences, which we think is essential here, in that the graph is a model of the inference process. As the inferences are being constructed, whether it be accomplished top-down or bottom-up, a chain of logical dependencies, between higher and lower level entities, is constructed and used to create explanations when needed for understanding and validation of the entities and their relationships.

(6) General purpose functional (bottom-up) and pattern based (top-down) inference algorithms (AI terminology "inference engines") suffer from severe performance issues, the main one being that they construct numerous non useful and/or invalid nodes during the process of constructing the useful and valid ones. Here, though, we don't need completely general purpose inference algorithms, constraints based on locality and time and functional category, and predictive filtering can be used to focus inference candidates, and encoding schemes which cluster data based on locality (e.g. quad tree location encoding) can significantly reduce the "search space" and lead to acceptably efficient inferences.

4. THE SYSTEM

4.1 BUILDING THE KNOWLEDGE BASE

First step involved in developing such a system is to build a knowledge base. Useful knowledge sources are hidden on the internet where the knowledge is often fragmented into partly visible information bits spread across various sites. The collection of scattered information and reconstruction of knowledge is a complicated task. The knowledge engineer needs to be fully aware of the problem domain and its culture in order to create a useful knowledge base. In the intelligence domain, the raw knowledge is either information or events. We gathered data for developing this knowledge base from open sources on the web/internet and converted the collected information into a meaningful facts or rules. We put the information and events into a standard, structured format as a source of data(facts) as we explain next.

All the facts in the prolog system are either an information fact or an event fact.

Both information and events are complex and are represented as lists which have the following attributes: info/event type, source, who, what, where, how, why, when.

In addition to the above attributes information and events have the nonfunctional attributes: Importance; Urgency; Pertinence(geographically and temporally); Reliability; and Credibility. The example below has **probability** and reliability representing these.

The attributes can also be complex and then also be represented lists, themselves. For example when as an attribute list, which has 3 elements namely month, date and year. Similarly where is also a list consists of 5 elements namely place_name, street, city, state and country. While we gather information if any of the attributes or elements is unknown the we substitute '-1' in its place indicating that the field is unknown(or irrelevant).

We make use of categorized lists with standard terms (ontologies) in the spirit of the ontology standardization community, or a dtd in XML. A few elements of the categorized list include air_terr, bio_terr, susp_hij, which means air terrorism, bio terrorism and suspected hijackers list respectively. In order to have a more realistic database, we also feed some random events and information that have nothing to do with terrorist threats. This is one of the options to check how the system behaves in the huge background of noises.

Examples illustrating the approach include:

(1) an example of information (facts):

%info(information_type, source, who, what, where,how, why, when, probability, reliability)

info(air_terr, fbi, terrorists, [train, pilot_training], ['aviation schools',-1,-1,-1,'USA'],-1, -1, [5,-1,1999],1,1).info(air_terr, fbi, terrorists, [train, pilot_training], ['aviation schools',-1,-1,-1,'USA'],-1, -1, [5,-1,1999],1,1).

(2) an example of information (facts):

% event (event_type, source, who, what, where, how,why, when)

event(arrest, media, ['Zacarias Moussaoui', flight_school_student], arrest, [-1,-1,'Minneapolis','MN','USA'] -1, immigration_charges,[08,15,2002])

(3) generalinformation:

% gen_info(info_type, [_,_,….]).

gen_info(places_where_terr_have_come_from, ['Afganistan', 'Saudi Arabia', 'Palestine', 'Iraq', 'Germany', 'Pakistan', 'Egypt', 'Syria']). %note this is more valid than "places_that_harbor _terrorists"

(4) examples of rules:

terr_attack(Type):- event(Type,_,_,_,_,_,_,_), (Type = terr_attack; Type = air_attack ; Type = chem_attack; Type = bio_attack) .

or, more complicated:

```
resources_necessary_for_hijacking(Cred):-
  ( info(air_terr,_,_,What,_,_,_,[_,_,YY],Pro,Cr),
  member(pilot_training, What),
  YY > 93,
  Pro > 3,
  Cr > 2 ,
  Cred = 2 );
  (event(_,_,Who,What,Where,_,_,[_,_,YY]),
  member(place_where_terr_from,Who),
  member(suspicious_behaviour,What),
  info(skills_acq_place,Source,_,_,_,_,_,_,_,_),
member(K,Source),
  member(K,Where),
  Cred = 2 ).
```

4.2 INFERENCING

Once the knowledge is represented in some form, the reasoning procedure draws conclusions from the knowledge base. There are 2 basic ways of reasoning:

Top down inference (Backward chaining) draws conclusions by asking important questions; In backward chaining we start with a hypothesis and reason backwards, in the inference network, towards the facts, it hopes are there to support a conclusion.

Bottom up inference (Forward chaining) draws all conclusions that are supported by the facts. In forward chaining the interpreter starts with what is already known, derives all conclusions that follow from this and adds the conclusions to the fact relation.

Which is better? Each has its advantages; our approach is to combine the both while doing the inferencing; we generally use bottom-up inferencing near the bottom and top-down near the top.

The rules we use for inferencing identify possible patterns or anomalies in the fact base. We use heuristics in making up those rules like "past predicts future". Our inference engine infers things from the knowledge base like a human intelligence analyst would infer naturally. As an example, The rule 'hijacking a plane', would be true with some probability if the facts 'resources necessary for hijacking' and 'Plane access' were true with some high probability. Again we can verify whether the rule 'resources necessary for hijacking' is true by looking the facts that people from the countries which are heaven for terrorism is learning flight school lessons in the US during the specific time period we are looking for and there is an intelligence warning obtained by the investigators that there might be a possibility of terrorist attack. This type of pattern is straightforward to encode, with decisions based on success of rules with inline calculation of probability.

4.3 THE ARCHITECTURE

The architecture will be described at two levels, the node architecture and the system architecture. The node architecture has to handle two functions, inferencing and communications(figure 4). objects with properties.

Inferencing's aim at the node level is to provide one level of abstraction in the event, object, situation, threat hierarchy. For instance the bottom nodes abstract raw facts into

The system architecture(figure 5) separates inferencing from communications and assigns inferencing activity compartmentalized by location, function, and/or organization hierarchy to limit the scope of inferencing. Its structure mirrors, as much as possible, the organizations structure with its dual, intelligence/control and functional specialty structurings. Information goes into the system at the place where it is detected. Alerts come out of the system at various, appropriate places. This architecture combines scalability, validation and appropriateness.

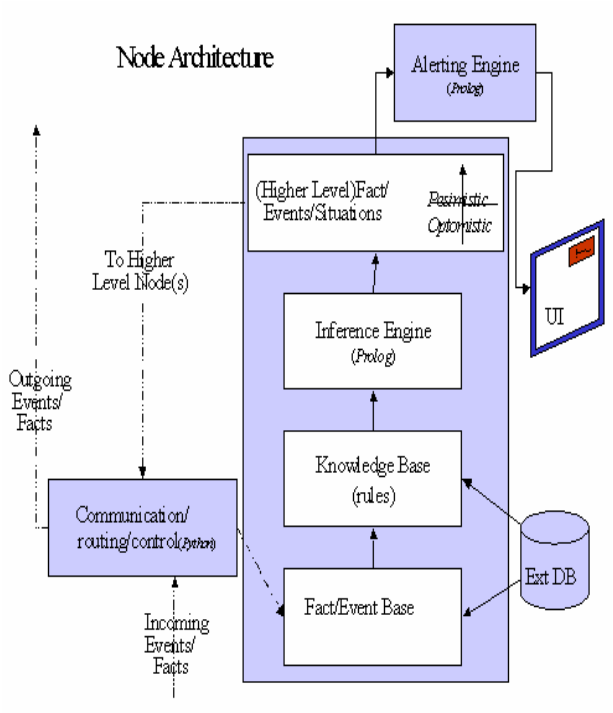


Fig. 4

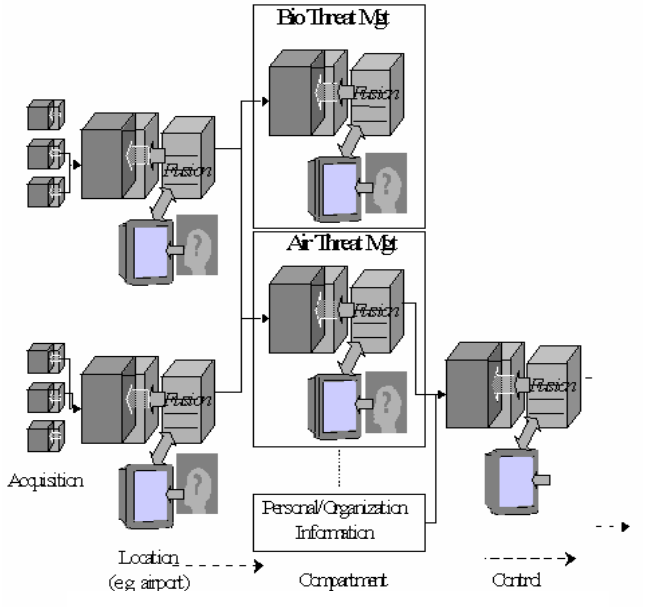


Fig. 5

5. ISSUES/EVALUATION

5.1 INFERENCING SCALABILITY

A major question arising in this type of systems is scalability/performance. We address this issue by fusing the data in different levels, and in different categories, along with spreading the inferencing across a distributed network of high performance computer clusters. In our system our first level rules are compartmentalized by location of the acquiring nodes. The next level is based on the type of terrorism threats (air terrorism, bio terrorism, chemical terrorism, other terrorism threats) and then in the upper level it is compartmentalized based on region. The communications network is intelligent, in that the communications protocols are aware of the structure of the next level up and thus filter prior to transmission, greatly improving performance. The hierarchical distributed cluster network reduces the performance bottlenecks by limiting the scope of information over which each node must inference.

The approach to the structure of fusion systems outlined here of abstracting "raw" data into meaningful, more abstract objects, events, courses of events, strategies, ..., based on organizing representations of these entities into layers, again limits the inferencing scope. Our hierarchy conceits of four "official" layers, info/events, objects, situations and threats, but additional abstractions structure each layer. Integrated with this layering scheme, entities are organized into a graph, usually a directed acyclic graph ("DAG"), adding a horizontal dimension (compartments) to the inferencing hierarchy.

Functional (bottom-up) and pattern based (top-down) inference algorithms suffer from a severe performance issue. They construct numerous non useful and/or invalid nodes during the process of constructing the useful and valid ones. (Note: bottom-up produces a preponderance of non useful inferences, top-down produces many non valid searches) The main problem is that computers have to try things, without knowing ahead of time which tries are going to be successful (valid and useful). Here, though, we don't need completely general purpose inference algorithms, constraints based on locality, time and functional category, and predictive filtering can be used to focus inference candidates, and encoding schemes which cluster data based on locality (e.g. quad tree location encoding) can significantly reduce the "search space" and lead to acceptably efficient inferences. In addition clustering is an effective way to allow parallel inferencing, significantly increasing fusion performance.

5.2 VALIDITY AND TRUSTWORTHINESS

We incorporate the validity and trustworthiness (and avoiding suspicion of politicizing) in inferencing by a number of mechanisms. First, we constraining the development of rules using validity fostering meta-rules like "past predicts future" and cause-effect. Second, we use an open source strategy to rule development with "red teaming" tests. We also overlay nonfunctional attribute (like importance and trustworthiness) calculation over the inferencing framework,

calculating and evaluating these attributes of the higher level nodes as properties or attributes of lower level nodes.

The last mechanism for validity of inferencing provides explanations to the user of how the inference was achieved. Once the system has come up with an answer to the user's question, the user should see the evidence: that is rules and subgoals from which the conclusion was reached. Such evidence consists of an annotated proof tree which can be represented in one of the following forms, depending on the case:

If P is a fact then the proof tree is P.

If P was derived using a rule if Cond then P then the proof tree is Proof <= Cond

If P is P1 **and** P2 then the proof tree is Proof1 **and** Proof2. If P is P1 **or** P2 then the proof tree is either Proof1 **or** Proof2

6. CONCLUSION

This paper addressed a major issue in intelligence by building a prototype of a valid inference engine with a scalable distributed architecture to facilitate fast, pertinent, valid information flow between agencies and locations, and to alert intelligence analysts of pieces of information that warrant their attention.

It described an approach to developing a scalable intelligence inferencing system for civilian terrorism intelligence. The system provides automated assistance in developing, and working with, a high level abstract view of existing and pending crisis situation. It provides assistance in planning and controlling the organization's operation in crisis situations. It provides visualization and conceptualization assistance at multiple levels, the object, situation, and operation levels. It provides for exploratory and intent developing reasons, and simulation (extrapolation) of future scenarios.

The approach addresses scalability, performance and validity in intelligence inferencing(fusion) by top-down and bottom-up inferencing, using a distributed network of computer clusters, compartmentalized by location, type of information and/or type of threat, fusing at progressive levels in layers of abstraction, incorporate validity and trustworthiness into inferencing, providing a flexible human interface including validity enhancing explanations.

7. REFERENCES

[1] I. Bratko. Prolog Programming for AI. Addison Wesley Publications. Third edition, 2001.

[2] W.F.Clocksin and C.S.Mellish. Programming in Prolog, 4th edition. Springer-Verlag publication.

[3] Hongge Gao. Formal Information Fusion Framework. Information fusion group, Dec 2000. http://www.ece.neu.edu/groups/ifg/projects.htm

[4] Terrorism in the United States. US department of Justice. FBI publications. 1999 Report. http://www.fbi.gov/publications/terror/terroris.htm

[5] Intelligence Monitoring. A special report from Answerchanse Inc, July 2001.

[6] D. Rozenstein, N. Minsky. Controlling the use and evolution of Database systems: A Prolog based approach. Journal of Management Information Systems, v.3 n.1, p.5-31, Summer 1986.

[7] Meeting the airport security challenge. Report of the Secretary's Rapid Response Team on Airport Security, Oct 2001. http://www.arsa.org/lobbying/ airport%20security.htm

[8] CIA Views on International Terrorism. DCI Testimony: 3/20/96, http://www.kimsoft.com/korea/ciachem1.htm

[9] US: FBI Chief Admits 911 Might Have Been Headed Off. New York Times. 30 May 2002. http://www.mapinc.org/drugnews/v02/n1016/a09.htm

[10] Could 9/11 Have Been Prevented?. Time online edition. Aug 2002. http://www.time.com/time/nation/article/0,8599,333835,00.html

[11] W. N. Grigg. Did We Know What Was Coming?. The new American, Vol.18 March 11, 2002.

[12] US Intelligence and Security Agencies. FAS. http://www.fas.org/irp/official.html

Thomas J. Wheeler *is an assistant professor at the University of Maine. He started at the University in the Fall of 1999 after a thirty year career as a Physicist, Electronic Engineer and Computer Scientist a US Government research laboratory. He received the Ph.D. Degree in 1988 from Stevens Institute of Technology in Hoboken NJ, directly across the river from New York City. He is currently working on Integrating Infrastructure for multidiscipline and Bio-Informatics Systems.*

Research Areas: Software Engineering, Design, Design Assistance/Automation, Environments, Programming Languages & Technology, Documentation Design and Technology,

Distributed Systems, Modeling and Simulation, Database Systems.

***Karpagavalli Vel** graduated with a Masters Degree in Computer Science in 2003. Her current work is in High Performance, Scalable Intellegence Inferencing Systems for Civilian Terrorism Intelligence. Her approach was to: Migrate Low Level Fusion/filtering Down to Sensors and Intelligence Network; Create Unified Spatial/semantic/temporal Database (Model) Infrastructure; Fuse at Progressively Higher Levels in Layers; Inferring Object(s), Situation(s), Organization, Intent and Threat(s), From Events and Information (Using Unified Top-down and Bottom-up Inferencing and Validation): Meet Performance Goals by Compartmentalizing Information and Inferencing; and Using Distributed Clusters for Inferencing; Incorporate Validity and Trustworthiness Into Inferencing and Create a User Interfacing Strategy With Perceptual Enhancements to Focus Users on Characteristics Which Interpret the Situation and Its Unfolding Effectively.*