HANDLING EVENTS IN INTELLIGENCE DOMAINS*

Lisa A. Jesse  
GTE Government Systems  
1450 Academy Park Loop  
Colorado Springs, CO 80910

Jugal K. Kalita  
Department of Computer Science  
University of Colorado  
Colorado Springs, CO 80933

ABSTRACT
In this paper we discuss an user-friendly system that aids users in analyzing temporal information in various intelligence domains. The analysis performs situation assessment, predicts future events, and explains decisions in an understandable graphical and textual description. The strength of the system lies in the well-seamed integration among the various modules and the maintainability of the knowledge base by computer naive users. It represents the new breed of current commercial systems which use AI techniques where it best suits, not just for the sake of AI.

INTRODUCTION
One of the primary responsibilities of an intelligence analyst is to keep decision makers apprised of the current situation and give projections of future activities. This process consists of monitoring observable events for indications of particular activities and deviations from expected norms. Events consist of activities such as communications, political activities, and aircraft maintenance.

Situation assessment and predictive analyses are very time consuming and require extensive training. Currently, analysts manually detect indications of particular activities by visually pattern matching events against general event templates. These event templates consist of key events and event relationships known only by the most experienced analysts.

There are several problems in situation assessment. The knowledge in intelligence domains is in continuous flux. Only the most experienced analysts have the knowledge required for accurate and timely analysis. Maintaining the appropriate level of knowledge is difficult because of the lack of documentation and frequent personnel turnover. Often, relevant event sequences are obscured by numerous irrelevant events. This "noise" hinders the analyst and may lead the less experienced analyst to wrong conclusions. In addition, key events may not have been detected or not have occurred.

SYSTEM OVERVIEW
The primary consideration in the design of the system is to provide flexibility in knowledge acquisition and modification. Specifically, the knowledge must be maintainable by users who are experts in their particular analysis domain but are relatively inexperienced in computer software. As a result, the knowledge representation must mirror how they manually perform situation assessment and predictive analysis. In addition, the expert system toolset should allow reusability of software across various domains.

As shown in Figure 1, the system is composed of three basic processing components: the Temporal Analysis System (TAS), the Model Developer, and the Knowledge-based Prediction Analysis and Situation Assessment (K-PASA) expert system. The Model Developer allows the user to maintain the expert system's knowledge base. Details of the system are discussed in [Jesse 93].

Figure 1. System Overview.

The primary input into K-PASA is a set of events selected by the user and a set of models from the database. The expert system returns descriptions of activities most likely indicated by the events. From these descriptions, the user can have the expert system predict future events. In addition to providing analysis capabilities, the expert system contains an explanation subsystem. This subsystem provides

*Substantial portion of the work reported in this paper was sponsored by Air Force Rome Laboratory.
transparency into system conclusions by using a combination of graphics and natural language text.

K-PASA and the Model Developer were originally developed on a VAXStation 3520 platform. The database was Rdb, a relational database from Digital Equipment Corporation. The user interface was developed using DECwindows and GKS graphics software. The Model Developer was implemented in the Ada programming language. The K-PASA system was also implemented using Ada, but the core functionality was developed using the NEXPERT OBJECT AI development environment. Since the original development, the tools have been ported to a DECSTATION/Unix/MOTIF environment using the Sybase DBMS.

KNOWLEDGE REPRESENTATION: TEMPORAL TRANSITION MODELS

As previously mentioned, knowledge in the intelligence analysis domain is in continuous flux. In order to maintain current knowledge and still be cost effective, the knowledge representation must be maintainable by an analyst who works with the system on a day to day basis. As a result, the system cannot require any specialized software training, let alone training in AI knowledge representation techniques.

There are many sophisticated models for the representation of time such as [Allen 91, Dean 89]. However, to provide knowledge acquisition efficiency and still maintain the appropriate representational power, a relatively simple, but elegant approach to knowledge acquisition was used in this project. This knowledge representation called Temporal Transition Models (TTMs) specifies generalized event patterns that characterize a particular activity of interest. Associated with the TTM is a graphical specification language developed to be consistent with the method by which users manually perform situation assessment and predictive analysis. This graphical language is supported by the Model Developer. The TTM representation together with the graphical language allow analysts to maintain the knowledge base without the aid of a knowledge engineer. For details beyond what is given here, see [Jesse 93].

TTM Structure

The TTM structure is a combination of concepts from Augmented Transition Networks (ATNs) used in natural language processing [Woods 70] and decision trees. Like ATNs, TTM are composed of states and transitions. State correspond to events in the application area (e.g. communications and personnel movements) and are graphically represented by circles with an icon denoting the event type in the center. Within the state specification, the user can specify particular events by defining constraints on event attributes.

Transitions define the ordering of the states and provide temporal constraints. The temporal constraints specify the time that the events must have occurred in order to satisfy the next state, relative to the event(s) fulfilling the previous state. Currently, the reference point used in evaluation is the event start times. Multiple transitions from a particular state are considered a branch. Transitions in a branch can either be "AND" or "OR" transitions. The evaluation of transitions is very similar to AND/OR decision trees, where OR branches are evaluated independently and AND branches are evaluated together.

Confidence values are associated with transitions. These values range from -100% to 100%, where negative values indicate a decrease in belief and positive values indicate an increase in belief. The confidence is additive across transitions traversed over a particular TTM path. Combination of confidences based upon the various paths satisfied by the events is based upon Mycin confidence factors [Luger 89]. More formal probabilistic methods were investigated, but were discarded as being too cumbersome for the analysts. The confidence values represent the level of belief that the events indicate the correlated activity as well as the accuracy of the TTM description. Generally, as more states are traversed during the analysis, the confidence that the activity is occurring increases. The method chosen accurately models the manual method used the analysts, thereby making it more natural and less threatening to the user.

The TTM are stored in a relational database for access by the Model Developer and K-PASA. Figure 2 describes tables that support the communication, aircraft and VIP movement event types.

Example

An example of a simple TTM describing Strategic Air Command (SAC) Operational Readiness Inspection (ORI) activity is shown in the top graphics pane in Figure 3. This figure is an explanation screen. SAC, which has been recently deactivated, periodically performed ORIs to exercise the combat readiness of its subordinate units. All of the unit and aircraft information was derived from a 1991 issue of Air Force Magazine in order to illustrate the concepts on the TTM representation. Four event types are illustrated in the TTM: codewords (square icon), airborne command posts (straight-wing aircraft icon), communications
describe the general meaning of the TTM and point out use of key concepts.

The leftmost codeword event is the initial state. This state describes General Strategic Alert activities by SAC headquarters. In detail, this state specifies that matching events must be codewords from SAC headquarters issued to the headquarters of the 15th and 8th Air Forces. From this initial state, two paths can be taken. The top branch describes a typical 8th Air Force exercise and the bottom branch describes a 15th Air Force exercise. The transitions in this branch are "OR" transitions so that either one or both Air Force entities may respond to the SAC ORI alert. Precursor 15th Air Force activities before the actual conduct of the exercise can be summarized as follows: the 15th Air Force contacts subordinate units that will participate in the exercise, a 15th Air Force airborne command post will be active to provide orders, participating units will confirm alert status, then the 15th Air Force will initiate the exercise. Depending on whether the aircraft activity is to be performed or simulated, aircraft will depart from various bases associated with the participating units and then perform bombing runs on the designated bombing areas. After the units have completed the live or simulated bombing runs, participating units will contact the 15th Air Force headquarters, followed
by the 15th Air Force headquarters contacting SAC. The 8th Air Force exercise progresses in a similar fashion, but with units subordinated to the 8th Air Force rather than the 15th Air Force.

Let us now consider the specifications of the transition to the first codeword state on the 15th Air Force branch. Although not shown on this figure, this transition specifies that 1 to 10 minutes after event(s) that match the initial state are detected, we should expect the occurrence of codeword event(s) issued from the 15th Air Force headquarters to one or more subordinate units. If event(s) which satisfy these constraints are found, the confidence that a SAC ORI is occurring is raised by the confidence increment defined in the transition. A variable has been defined for the subscriber’s attribute in the state specification. Semantically, this variable captures all of the subordinate units participating in the exercise and is used in subsequent states to further constrain events associated with the participating units. The Model Developer allows the user to examine the specifications of the transitions and states, however space constraints prohibit a separate figure.

**K-PASA SITUATION ASSESSMENT**

Situation assessment involves the detection of key activities from incoming event descriptions. Research in event-based situation assessment and event prediction has been active in the field of political science for several years. Early work in this area included a weight-based method of comparing event sequences called the Levesehtein Metric [Mefford 84]. More recent work includes POES [Heise 88] and situation predisposition models [Hudson 91].

For situation assessment, the primary input into K-PASA is TAS events and a set of TTM's describing the correlation between events and activities of interest. K-PASA is very easy to use. Currently, K-PASA is invoked via a TAS Timeline where the user is asked to select the events to be processed. The initial K-PASA screen contains the types of activities (e.g., the activities with defined TTM's) known to the system. A status line is provided to guide the beginning user. The user may then select the activities to be detected by the system. K-PASA then performs the mapping between the events and the TTM's associated with the selected activities, returning the assessments listed in order of decreasing confidence. Assessment display thresholds can be set by the user. If the user selects an assessment, the confidence will be displayed. At this point, the user may request an assessment explanation or a prediction of future events.

**Mapping of Events**

The comparison process starts at a TTM's initial states, searching for events that match the initial states' specifications. If one or more initial states are satisfied, then the system searched for events that match the constraints specified by subsequent transitions and states. This traversal process continues until all TTM branches terminate or no events satisfy next transition/state specifications.

The building blocks in the situation assessment process are *hypotheses*. Each assessment has an associated set of hypotheses which describe aspects of distinct mappings between the events and TTM. Specifically, the hypotheses contain a list of traversed states, matching events and a confidence. For a given assessment, hypotheses describing the assessment are organized into a hierarchical structure. One assessment corresponds to one hypothesis hierarchy. The “root” hypotheses are generated for initial states. Children hypotheses are created when the comparison process encounters a branch in the TTM.

After all of the TTM's have been processed, the system aggregates the hypothesis hierarchies to create assessment specifications. This aggregation process collapses the hypothesis hierarchy into a single structure and determines the final confidence of the assessment. Associated with each hypothesis is a confidence. During aggregation, the confidence in the assessment conclusion is determined from the branch types using the standard confidence combination theory used in MYCIN. The MYCIN combination rules are: $C_1$ or $C_2 = \text{MAX}(C_1,C_2)$; $C_1$ and $C_2 = \text{MIN}(C_1,C_2)$.

As the data gathered in the intelligence environment is not perfect, events may not have been entered into the system exactly according to state specifications. One particular problem is that multiple events combined together may satisfy a state. Instead of being entered into the system as a single event, they were entered as individual events. K-PASA compensates for this problem by searching for and combining events that together satisfy a particular state. These multiple events contribute to the creation of a meta-event. A related problem is that an event may be encapsulated in a larger event. For example, an event in a communications event may encapsulate a single source talking to multiple destinations. The system will also parse large events to find embedded events. Because events that do not satisfy TTM specifications are ignored, other pattern matching problems such as irrelevant event noise are inherently solved by the system.
ASSESSMENT EXPLANATION

Due to the critical nature of their work, the analysts need to understand and accept the reasoning behind system conclusions. Figure 3 shows a typical K-PASA assessment explanation screen. The explanation is a mixture of graphics and natural language text. In the top pane, graphics illustrate the TTM structure with traversed states filled for easy identification. Matched events are displayed in the middle pane using a timeline display. The graphics allow for quick, superficial explanation in situations where the analyst is pressed for time. The natural language text, displayed in the bottom pane, provides explanation details when the user has the time and inclination for more in-depth explanation. The text communicates the reasoning behind the assessment without overwhelming the user with irrelevant details.

The work in the translation of formal proofs into natural English text by Chestor [Chester 76] has ideas directly applicable to the explanation subsystem. We have also used ideas in [Kalita 88] and [Allen 87].

Text Generation

The overall structure of the explanation text is divided into two sections. The first section consists of an introductory paragraph which provides a high level summary of the degree of matching between the TTM and the input events. In particular, this paragraph informs the user of the type of phenomena that was detected and the confidence the system has in the conclusion. In addition, the main portions of the TTM that were satisfied are summarized using relationship specifications in the states. These relationships describe the meaning of individual states or groups of states to the activity being described in the TTM.

The second section contains the main body of the explanation which describes in detail the relevant events that matched the TTM. This multi-paragraph section describes the relevant events that matched the TTM and is structured according the time-ordered relationships in the TTM. The main body is composed of one paragraph for each relationship listed in the introductory paragraph. In each paragraph, sentences are generated that describe the event(s) that matched the states within the relationship.

During the planning process, a graph is constructed using the relationships in the TTM as a general outline. Nodes in the graph consist of sentence specifications and correspond to one sentence in the text. These nodes are ordered in time and therefore in presentation order. The specifications include information concerning events to be mentioned in the sentence, how the events are related, and the satisfied states. In particular, there are two types of nodes created: main body topic nodes and main body sentence nodes. A sample graph is shown in Figure 4. The main body topic nodes organize the main body paragraphs according to the state relationship specifications. These nodes contain information about the main body sentence nodes in the paragraph as well as information required for generating the paragraph's topic sentence. The main body sentence node specifications include information concerning events to be mentioned in the sentence and how the events are related. Multiple events which in conjunction satisfy a single state are combined into one main body sentence node.

Figure 4. Sentence Specification Graph.

The introductory paragraph is simply a set of preconstructed sentences with slots being filled in by the appropriate objects. This is appropriate as the introduction always explains the same features of the hypothesis and these features are always available.

The construction of the main body paragraphs is more complicated. Basically, the text generation process traverses the graph generated by the planning component to generate the main body paragraphs. A realization rule is defined for each type of event and the paragraph introductory sentence. To focus on important event attributes rather than reciting all attributes, the event realization rules describe the event in terms of its relevant and key attributes. An attribute’s relevancy is determined by the state matched by the event. If the state places a constraint on a particular event attribute, then that attribute is mentioned in the sentence. Key attributes are always mentioned. An example aircraft event realization rule is described below.

Aircraft Event Realization Rule =
[relative time adj][<number>][<unit>]<type>
type phrase <activity> [location prep<location>][heading verb][heading>]

Objects in [] are associated with relevant attributes or optional words. Objects in <> are slots filled in by
the event(s) attribute values. The objects in italics are words and phrases retrieved from the lexicon.

Attributes of multiple events within a single node and attributes with more than one value are combined with an "and" inserted at the appropriate spot. Explicit event times are also not mentioned. Instead, adjectives describing the relative timing between events (e.g., first, last, next) are placed at the beginning of the main body sentences when appropriate.

The explanation text seen in the bottom pane of Figure 3 is:

**THE EVENTS SEEM TO INDICATE SAC OPERATIONAL READINESS INSPECTION ACTIVITY WITH A CONFIDENCE OF 1.00. THIS PHENOMENON IS DESCRIBED IN MODEL NUMBER 335 DISPLAYED ABOVE. IN PARTICULAR, THIS IS INDICATED BY GENERAL STRATEGIC ALERT, BATTLE STAFF ACTIVATION, CONFIRMATION OF ALERT STATUS, EXERCISE INITIATION, LIVE PLAY AND EXERCISE TERMINATION BY SAC.**

**GENERAL STRATEGIC ALERT IS INDICATED BY THE FOLLOWING EVENTS. FIRST, A HQSAC CODEWORD WAS DETECTED WITH SUBSCRIBERS HQ8AF AND HQ15AF. NEXT, A HQ15AF CODEWORD WAS DETECTED WITH SUBSCRIBERS 57AD, 92BW, 22ARW, 93BW, AND 384BW. LAST, A HQ8AF CODEWORD WAS DETECTED WITH SUBSCRIBERS 12AD, 65ARW, 7BW, AND 509 BW.**

**BATTLE STAFF ACTIVATION IS INDICATED BY THE FOLLOWING EVENTS. COMMUNICATIONS WERE TRANSMITTED FROM 96BW, 22ARW, 384BW, 92BW, 93BW AND 57 AD TO HQ15AF.**

**EXERCISE INITIATION IS INDICATED BY THE FOLLOWING EVENT. A HQ15AF CODEWORD WAS DETECTED WITH SUBSCRIBERS 57AD, 92BW, 22ARW, 93BW, 96BW, AND 384BW.**

**LIVE PLAY IS INDICATED BY THE FOLLOWING EVENTS. FIRST, B-52, KC-135 AND B-1 WERE DETECTED DEPARTING AT MINOT AFB, DYESS AFB, FORDLAND AFB AND MCGILL AFB. NEXT, B-1 WAS DETECTED ENROUTE AT LAHUNTA CO.**

**EXERCISE TERMINATION IS INDICATED BY THE FOLLOWING EVENTS. FIRST A HQ15AF CODEWORD WAS DETECTED WITH SUBSCRIBERS 57AD, 92BW, 96BW, 93BW, 22ARW AND 384BW. NEXT, A HQ15AF CODEWORD WAS DETECTED WITH SUBSCRIBER HQSAC.**

**PREDICTIVE ANALYSIS**

At a high level, the situation assessment is a two phase process: 1) monitoring observable events for indications of particular activities, then 2) based upon those assessments predict future activity. The KPASA predictive analysis component predicts future events using information from a given assessment and the associated TTM. The predictive analysis component is invoked by the user selecting an assessment generated by the situation assessment component and providing a prediction timeframe.

KPASA predictive analysis processing is relatively straightforward. Basically, the system predicts next events by looking at the states yet to be fulfilled. Paths stemming from the last states matched in the assessment’s hypothesis hierarchy are analyzed by the system using the event(s) matching those last states as time references. As in the situation assessment component, event start times are the reference points in the evaluation. For example, in Figure 5, the shaded circles denote states traversed in a particular hypothesis hierarchy. The start time of the event(s) that matched state 2 is used as a reference time for predicting events associated with subsequent states in branch 1 (e.g., states 3, 4, 5).

The start time of event(s) that matched state 7 is used as a reference time for predicting events associated with subsequent states in branch 2 (e.g., states 8 and 9). This reference time is propagated through the TTM transitions until the timing of the predicted events exceeds the prediction timeframe specified by the user or no more states exist.

![Figure 5. Prediction Example.](image)

The constraints in the state specification provide additional information on predicted event attributes. For example, assume that state 3 specifies a communication event with the source equal to SAC Headquarters and the destination equal to either the headquarters of the 15th or 8th Air Force. These attribute constraints on the source and destination attributes are used to further define the predicted communications event.

Like assessment explanation, the prediction display also uses graphics and text. The TTM associated with the assessment is displayed with the
states traversed filled and the predicted events highlighted in yellow. Textual description of the predicted events is displayed at the bottom of the screen. The events are ordered in increasing predicted start time ordered. As the discussion of predicted events is relatively trivial compared to the hypothesis explanation, the predicted event text generation does not utilize any sophisticated natural language generation techniques.

**TTM ACQUISITION TOOL**

To ease the knowledge acquisition bottleneck, a knowledge acquisition tool, called the Model Developer, was developed to allow domain experts to enter and maintain the knowledge without the aid of a knowledge engineer. The Model Developer allows the user to add, modify, and delete TTM's using the TTM graphical specification language.

The manipulation of the TTM's is straightforward. The Model Developer maximizes the use of mouse point-and-click object manipulation and X-Windows capabilities. Pull down menus appear at the top of the Model Developer window. The specifications of both states and transitions are form-based for easy data entry.

Because of the requirements to archive and document the knowledge contained in the TTM's, a powerful annotation subsystem was incorporated into the Model Developer. This annotation subsystem provides the ability to annotate the TTM's with comments and other graphical annotations (e.g., lines, circles, arcs, polygons). Annotations are very useful in providing additional explanation or pointing out important information. This capability is very useful when creating briefing slides or using the TTM's for training.

To support operational TTM development, the Model Developer supports two types of TTM's: working and validated. Working TTM's are considered models "in progress". These TTM's are not validated and are not saved in the database for processing by K-PASA. Validated TTM's, on the other hand, have been validated by the system to be syntactically correct and are stored in the database. Examples of validation include precluding the user from defining only one AND transition in a branch. Typically, TTM's are in the working mode until the analyst feels comfortable in using the TTM in the situation assessment processing.

**APPLICATION AREA PORTABILITY**

A primary consideration in the development of K-PASA and the Model Developer was to design the tools to be relatively domain independent. Providing application area portability allows the tools to be customized to a particular analysis topic (e.g., from command and control to counter terrorism) with minimal effort.

Application area portability is supported by an architecture which divides the processing into two levels: an application area independent kernel level and an application area specific application layer. In K-PASA, the kernel performs a comparison between the input events and the TTM's. The processing is general and does not presume anything about the contents of the TTM's or the events. K-PASA application layers define the particular application area in which the expert system tool will operate. At a minimum, the application layer defines the types of events to be processed by the kernel. The application layer development includes specific event type related processing such as database queries to the appropriate TTM and event database tables, comparison of event attributes to state specification. Typically, the application layer is less than 15% of the total code.

The application layer development for Model Developer also requires definition of the appropriate event types. This definition process requires developing the queries to the TTM database tables and creating the event-specific user interface for the state specification forms. To date, the user interface development has only required one DECwindows/MOTIF User Interface Language (UIL) file and one Ada package for each event type. The Model Developer uses the same TTM tables accessed by K-PASA. However, as the Model Developer adds, modifies and deletes TTM database records, the TTM database queries required for the Model Developer are a superset of those required for K-PASA.

This methodology has been used to not only add new event types to an existing application area, but has also been used to customize the tools to a completely new application area. For the average event type (around six attributes), the development time is approximately five hours.

**CONCLUSIONS**

This paper presents an approach for applying knowledge based systems technology to the problem of situation assessment and event prediction. Due to the requirements for intensive knowledge maintenance, traditional expert system knowledge representation and acquisition methods are not viable solutions. As a result, a new type of knowledge representation was developed that allows analysts who are domain experts, but have limited computer skills, to design and manipulate the knowledge. This
representation, called Temporal Transition Models (TTMs), describes generalized sequences of events expected to characterize a particular activity. Associated with TTM is a graphical specification language that corresponds to the methods in which the analysts manually perform situation assessment and event prediction.

The system, customized for command and control, was installed in December 1991 for evaluation. Since that time, the system has gone operational and is used on a daily basis. The tools have been customized to other domains and are currently installed in various locations. User response to the knowledge representation and system capabilities has been very positive. The close mapping between the manual timeline analysis methods and the graphical TTM specification language supported in the Model Developer provide for a truly user maintainable system. In addition to utilizing K-PASA to aid in analysis tasks, the analysts also use the system and TTM for knowledge archival and training. Because the TTM formalize the techniques known only by the senior analysts, the TTM and K-PASA allow for an easier transition of knowledge to less experienced analysts. Because each analyst can have their own set of TTM, K-PASA can be used to verify theories about how situations unfold.

From user feedback, many desired enhancements to K-PASA and the Model Developer have been identified. A near term enhancement (Summer 1993) will be the development of a real time capability. In real time mode, K-PASA will run continuously in the background analyzing events as they are entered into the TAS database. As significant activities are discovered, the user is notified by alerts. Other enhancements will include exploring the use of more sophisticated temporal models (such as [Allen 87, Dean 89]), adding a NOT constraint (e.g., looking for the absence of an event), and detection of missing key events, among others.

REFERENCES


[4] Dean, T., Firby, J., and Miller, D.


