Towards an Architecture for Reasoning about Complex Event-Based Dynamic Situations

Gabriel Jakobson, John Buford
Altusys Corp, USA
jakobson,buford@altusys.com

Lundy Lewis
Southern New Hampshire University, USA
l.lewis@snhu.edu

Abstract

In this paper we are concerned with event-based situation analysis. Application areas include the understanding and awareness of complex unfolding scenarios such as homeland security threats and future battlespace engagements. The paper (i) discusses the differences between the environments/requirements for event-based management and situation management, (ii) presents an argument for an integration of an event correlation system and a situation awareness system, and (iii) proposes an integrated architecture that combines rule-based spatio-temporal event correlation and case-based reasoning for understanding and managing situations. In addition, the paper discusses the hard problem of the mutual influence between event management and situation awareness.

1. Introduction

Consider that one is driving an automobile down a country road. What does one attend to mentally when driving on a country road? One might be enjoying the scenery, listening to music, or simply cruising. However, if a deer runs across the road in front of the car then the situation changes. For a brief second, a new kind of situation presents itself – an unsettling event occurs that must be dealt with immediately. After the event is dealt with, where hopefully the deer and car don't collide, the situation returns to normal.

Consider now that the country road leads into a four-lane street with the hustle and bustle of shops, restaurants, malls, stoplights, and heavy traffic. The driving situation changes once again. The driver attends to events such as changes in stoplights, brake lights, pedestrians, 491 Amherst St., the cheapest gas prices, and the like. Further, if one hears a siren nearby, one is likely to pull over – that is, the situation changes again.

The salient features of our driving story are (i) observations of events, (ii) dynamic situation awareness, and importantly (iii) the mutual influences between event observations and situation awareness. In the driving story, for example, it is easy to see that certain observations of events invoke certain situation templates, while at the same time certain situation templates lead to the expectation of certain event observations.

The focus of this paper is on analysis of dynamic situations. In particular, we are concerned with situations such as those encountered in the management of a battlespace, complex technological systems and processes, and real-time emergency situations in health care, homeland security, and other applications. These applications involve a large number of dynamic objects that change state in time and engage each other into fairly complex spatio-temporal relations. From the management viewpoint it is important to understand the situations in which these objects participate, to recognize emerging trends and potential threats, and to undertake protective actions that lead to predefined safe situations.

For example, in a tactical land/air battlespace, the operational units (troop formations, vehicles, weapon systems) maybe be under attack by multiple enemy sources, and thus the commanders need to know the changes in the battlespace situation, the direction, and the strength and severity of potential enemy actions [1]. Understanding dynamic battlefield situations is not a trivial task: it requires complex cognitive modeling, i.e. modeling the semantics of event perception, situation comprehension, and action projection [2]. These tasks should be undertaken in a dynamic environment, where events, actions, and situations follow the time constraints of the domain and the logic of temporal reasoning.

Modeling dynamic situations has been the research focus of several scientific disciplines, including operations research, ergonomics, psychology, and artificial intelligence. Most notably, John McCarthy introduced the notion of Situation Calculus in 1963, and several years later it was formalized by McCarthy and Patrick Hayes [3]. Informally, situations were considered as snapshots of the world at some time instant, while a strict formal theory was based on non-monotonic reasoning. Reiter proposed a formal situation calculus designed for action planning purposes, where situations

were defined as a sequence of actions [4]. Reiter's situation calculus has been applied in areas such as agent programming and robotics [5,6]. Reiter's situation calculus has been extended with a fluent-based (i.e., situation dependent actions) approach to model long-term, autonomous services in ubiquitous telecommunication applications [7]. Several other approaches to dynamic situation modeling have been studied, including approaches based on Petri nets, finite state machines, and Bayesian networks [8].

Our approach to modeling and reasoning about dynamic situations is inspired by two AI disciplines in which the authors have been involved for several years: Real-Time Event Correlation (EC) [9] and Case-Based Reasoning (CBR) [10]. EC is a widely recognized approach for telecommunication network root cause fault analysis and CBR is an effective paradigm for reasoning and decision support in applications such as health care, diagnostics, and legal reasoning. In this approach we propose the paradigm of Dynamic Case Based Reasoning, where the notion of classical cases used in CBR is extended by dynamic capabilities based on EC technology. The dynamic cases are used as the fundamental units of constructing complex situations, where the dynamics of situations is driven by the events and event correlation functions, as shown in Figure 1.

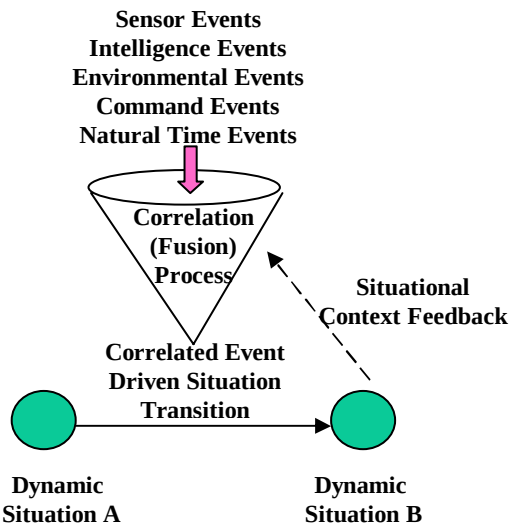

Figure 1. Correlated events and dynamic situations

Section 2 discusses the different features of dynamic event-based situation awareness and the traditional tasks of event management in communications networks. Section 3 discusses a conceptual framework for reasoning about situations. Section 4 discusses a design based on the integration of event correlation and case-based reasoning, and section 5 provides a conclusion and direction for future research.

2. Modeling complex event-based situations

Reasoning about complex event-based dynamic situations differs from the traditional task of analyzing a stream of events caused by faults in communications networks. For example, traditional event management systems are characterized as follows:

- The operational environment is formed from a large number, 100s to 1000s, of interconnected elements.
- The topology of the operational environment is defined by two main structural relations: connectivity and containment.
- The class relations are formed by studying information that is given or derivable from vendor documentation.
- The operational environment is largely static or with minor changes during the event analysis process.
- Network events are well-defined structured text messages with limited syntactic variations; heterogeneity of the events results from syntactic sugar-coding of internal vendor policies.
- There are strong causal relations between the events due to the propagation of faults through inter-connected network components.

The objectives of network event management include root-cause analysis, discovery of network performance degradations, and re-routing network traffic. Those tasks often require an execution of a relatively long chain of simple rules. Each step involves limited semantic processing and could be implemented by a rule-based system. A certain amount of temporal reasoning is required, but this is not a dominant feature. As a rule the events are fast flowing and require very fast processing -- during critical events this might reach processing several hundred to a thousand events per second.

In contrast, dynamic situation management such as battlespace management can be characterized thusly:

- The operational environment is formed by a medium number (hundreds, rarely thousands) of inter-connected elements.
- The operational environment is defined by complex temporal, spatial, and domain specific relations.
- The class relations are formed by multiple complex ontologies, which are usually ill defined.
- The operational environment is highly dynamic, and often unpredictable.
- Operational events are very diverse in nature and modality, involving signal, textual, visual and other types of information; a high level of heterogeneity is the norm; and the fusion of data, information, and knowledge is a required component of the operational space situation analysis.

- Causal relations are typically weaker (compared with network events); however there are very strong temporal and spatial relations.

The overall objectives in situation awareness are to understand the complexity of dynamic situations, discover potential developments, analyze potential threats and catastrophic situations, and undertake actions which avoid undesired situations. The realization of those tasks requires an execution of a relatively small number of complex reasoning steps involving the recognition of semantically rich event patterns.

The task of situation representation is a constructive task based on the principles of object-orientation and domain ontology [11]. Before defining the situations, let's introduce an object state as a set of object parameter values. Following the ideas of McCarthy, we consider situations as states, which have an assigned time value, either a point or an interval time. Time is the critical distinguishing factor between states and situations, e.g. two identical states with different time values represent two different situations. Considering dynamic situations, we are interested not only in the state value at some particular time, but also in the nature of situation changes, their speed, and directions. The dynamics of situations is reflected by state transitions, i.e. movement of an object from a state to state. Theoretically, it is possible to use the model of Finite State Machines (FSM) to describe these transitions; however, the simplicity of state specifications and augmentation of transitions with simple input/output variables make the FSM approach ineffective. Using cases for describing situations and using correlated events for determining situation transitions provides a more powerful tool for defining the dynamics of the situation changes.

The fundamental unit for representing a situation is a dynamic case: the smallest conceptual unit, which has a closed and semantically independent meaning in a domain. A tank at a specific location performing a specific task at a specific time is a situation.

Each situation is represented by its object specification, including:

- Identification
- Class specification
- Slots (parameters)
- Slot attributes
- Predicates and relations over the slot values
- Actions
- Time

In the following discussion we consider situations and cases as synonyms.

Abstract situation classes are organized into hierarchies according to the domain ontology. As usual, the specific situations are constructed by parametric instantiations of abstract situation classes. It will be unrealistic to represent complex situations with a single case. The representation of complex situations relies heavily on combination of situations using relations, e.g.:

- Spatial relations: s1 LOCATED_AT s2, s1 ABOVE s2, s1 CO-LOCATED s2, s1 NEAR s2
- Administrative relations: s1 SUBORDINATE_TO s2, s1 DIRECTS s2
- Structural relations: s1 PART_OF s2, s1 CONNECTED_TO s2
- Other domain specific relations

While modeling dynamic situations, certain situations are identified as the start, target, undesirable, and transitional situations. One of the tasks, e.g. in dynamic battle-space situation modeling, is the identification of enemy threats and actions to avoid catastrophic or reach winning situations. The threats are considered as potential enemy hostile actions. In real situations the threats could be forces of nature, or unintentional actions.

An important component of transition of dynamic situations is the definition of the conditions of transitions. As discussed earlier, we will use event correlation technology to define these transitions.

3. A conceptual framework for reasoning about situations

Figure 2 shows the conceptual framework for reasoning about situations. The Dynamic Situation Model (DSM) represents the top-level model when one reasons about events and situations. To stress our point, consider the task of battle-space operations modeling. We distinguish between three levels of modeling of events and situations: the Operational Space Level, the Network Communication Level and the System Service Level. On the Operations Space Level the DSM describes the battle-space objects, relations, events, actions, and situations as described in Section 2. The dynamics of changes of situations is defined by event correlation functions over the multiple sources of sensor, intelligence, security and other information sources, as well operations management events such as commands and actions.

The actual information flow between different battle-space operational entities, such as humans, human groups, vehicles, weapon systems, infrastructure elements and other entities is conducted over the battlespace data communication networks. The Network Communication Level deals with Network Event Models that describe the traditional tasks of network fault, performance, and traffic event management. While modeling the battlefield situations, reasoning about potential threats, and planning battlespace management commands and

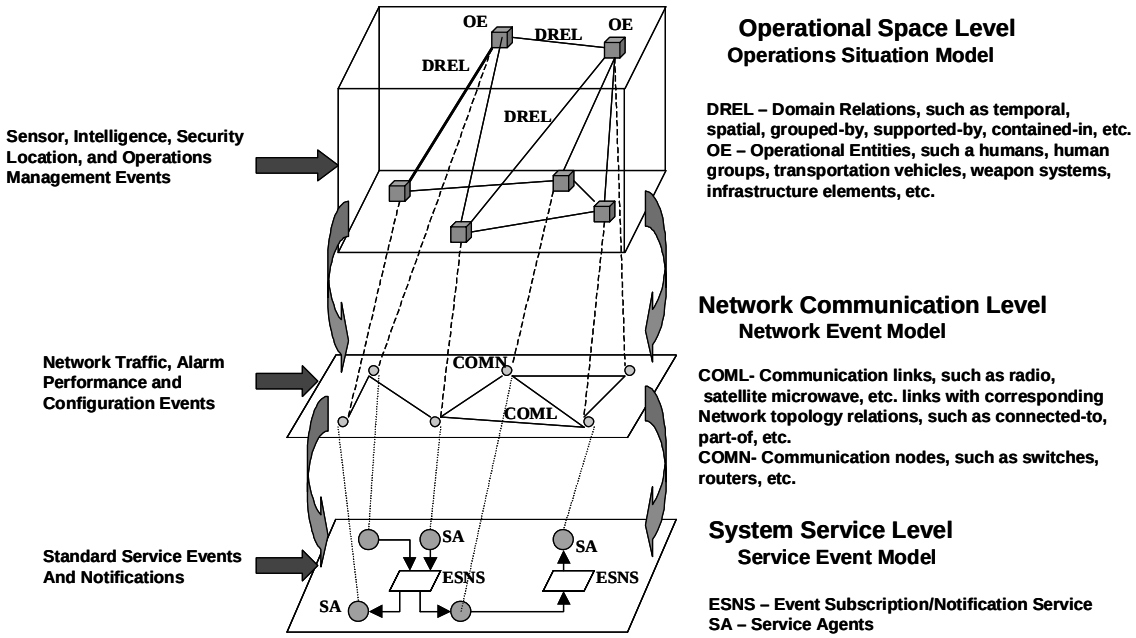

Figure 2. Modeling of dynamic event-based systems

actions, an important task is to map the events associated with the dynamic situation model to the events occurring in the corresponding network communications infrastructure. The overall task of such mapping is out of scope of this paper; however, we note that this mapping task is similar to the task of mapping events from Virtual Private Networks (VPNs) to the actual physical infrastructure networks.

The lowest level, the System Service Level, deals with software infrastructure objects, services, and events, which comprise the implementation architecture of the battlespace management platform. An example of such architecture has been described in [12]. It is built from standard CORBA services using structured CORBA events and the CORBA Event Notification Service. The CORBA Event Notification service was used as an event subscription mechanism and as a pipe to construct complex event passing channels.

4. Dynamic event-based situation awareness

4.1. Temporal Event Correlation

Event Correlation as a branch of Computer Science and Information Technology has been the focus of extensive research and practical applications over the last 15 years. The impetus for Event Correlation research was motivated by the fact that very often complex systems, being under operational stress, with malfunctioning of internal components, or being targets of malicious attacks, produce a large number alarms, which analyzed independently without recognizing the synergy between multiple information sources, do not reveal the actual internal situation of the system. In addition, large numbers of generated alarms might form chains of causally dependent events and mask the true root cause of the system failure. Due to the high speed of incoming events, the alarms may pass unnoticed or be noticed too late. Failure to capture the sequences of time-dependent events makes it hard to see the trends in the system of internal processes and predict the future behaviors of the system.

We will follow the event correlation model described in [9], where event correlation is broadly defined as a conceptual interpretation procedure of assigning a new meaning to a set of events that happen within a predefined time interval. This conceptual interpretation procedure stretches from a trivial task of event compression to a complex dynamic pattern-matching task involving Boolean functions, temporal relations and testing connectivity, containment and other structural relations between the objects that generate events.

In broad terms, an event is an act of internal transition of a system from a state to state, or in our area of interest – from a situation to a situation. The external manifestation of such an event is informational and as such it is an artifact created for human interpretation and practical utility. In more practical connotation, we consider an informational event as a time-stamped dynamic piece of information, which represents a change in the state of an object or manifests an action. Relative to the correlation process, we make a distinction between

the raw (i.e. base) events and the derived (i.e. correlated) events. The raw events are external events originating outside the correlation process, while the derived events are results of a correlation process. The overall event correlation process is run by the Correlation Engine, which uses the structural, spatial and other constraints defined by the objects in the situation model, and the temporal relations existing between the events.

Temporal relations between events, such as x AFTER y, x ENDS_BEFORE y, x SIMULANEOUS_TO y, play a critical role in event correlation applied to dynamic situation analysis. Some classes of temporal event correlation in an interval time have been discussed in [9].

Each event correlation process has an assigned correlation time window -- a maximum time interval during which the component events should happen. The correlation process is started upon the arrival of the first component event and stopped as the last component event arrives. As any other event, correlation has its time of origination, time of termination, and lifespan. By definition, the time of origination of the correlation is equal to the time of origination of the last component event. Event correlation is a dynamic process, so the arrival of any event instantiates a new correlation time window for some correlation. Time is a critical factor in the correlation process. First, very often the correlation processes should follow "event floods", which might reach hundreds if not thousands events per second. Second, event correlation patterns should take into account temporal orders and relations between events. In addition, the physical latencies in the communication lines could distort the actual order of incoming events causing incorrect pattern matching.

4.2. A CBR interpretation of situations

What is needed is a way to model dynamic situations. Further, the model should allow (i) learning from experience and mistakes and (ii) adapting more-or-less classic standard situations to accommodate the nuances of current situations.

Our approach to these tasks is to use a model of cognition, case-based reasoning (CBR), where a case is a template for some generic situation [10]. A set of events is posed to the CBR system, whereupon four processes are carried out. First, the set of events is compared to a library of situation templates, and a set of maximally similar cases is retrieved by a Retrieve Module. In the CBR literature, a number of retrieval algorithms have been proposed. The simplest and weakest algorithm is key-term matching; the most complex but strongest algorithm is analogy-based matching [13, 14].

The case library can be thought of as a set of former experiences with situations that are potentially similar to the situation at hand. Typically a former situation has to be tweaked in some way to render it applicable to the nuances of a current situation. This is the task of an Adapt Module. In the CBR literature, a number of adaptation algorithms likewise have been proposed. Null adaptation, for example, covers those episodes wherein a past situation is exactly like a current situation; adaptation by substitution covers those episodes in which an object that occurs as a descriptor in the current situation should be substituted throughout for an object that occurs as a descriptor in the retrieved case [13].

An Execute Module is straightforward. The user may choose to execute a command or action recommended by the retrieved/adapted case. The execution may be conducted manually or may be carried out automatically by the decision-maker, either in supervised or unsupervised mode. The execution of an action or plan may involve cooperation with other individuals.

Importantly, the results of the execution are recorded in the case and the case is entered back into the case library. In most CBR systems, the case library is structured as a sequential list, much like a stack of paper forms. Of course, decision-makers do not structure their problem-solving knowledge in this way. There have been several proposals for more complex memory structures in the literature. An interesting proposal is the concept of a master case [15]. A master case is one in which all the problem-solving experiences with a particular, well-defined situation are subsumed in one case. This is in contrast with the sequential memory in which each problem-solving experience is confined to a unique case.

4.3. An integrated EC/CBR architecture

Figure 3 shows an integrated architecture with an event correlation engine at the front end and a CBR engine on the back end. Based on the preceding sections, the design of the conceptual architecture is straightforward. The new feature that Figure 3 emphasizes is the flow of information in both directions between the correlation engine and the CBR engine. The events issuing from the correlation engine are used to invoke cases, or situation templates, that serve as interpretations of multiple events. Figure 3 suggests two cases that might be hypotheses about a current situation. In reverse direction, a case might suggest further information which, if it were available, would strengthen a hypothesis. The CBR engine could send a message to the correlation engine whereupon the engine attempts to prove the truth or falsity of a proposition or try to find the value of a parameter. If that fails, the CBR engine could alert the command center of an opportunity for seeking out information that would strengthen a hypothesis.

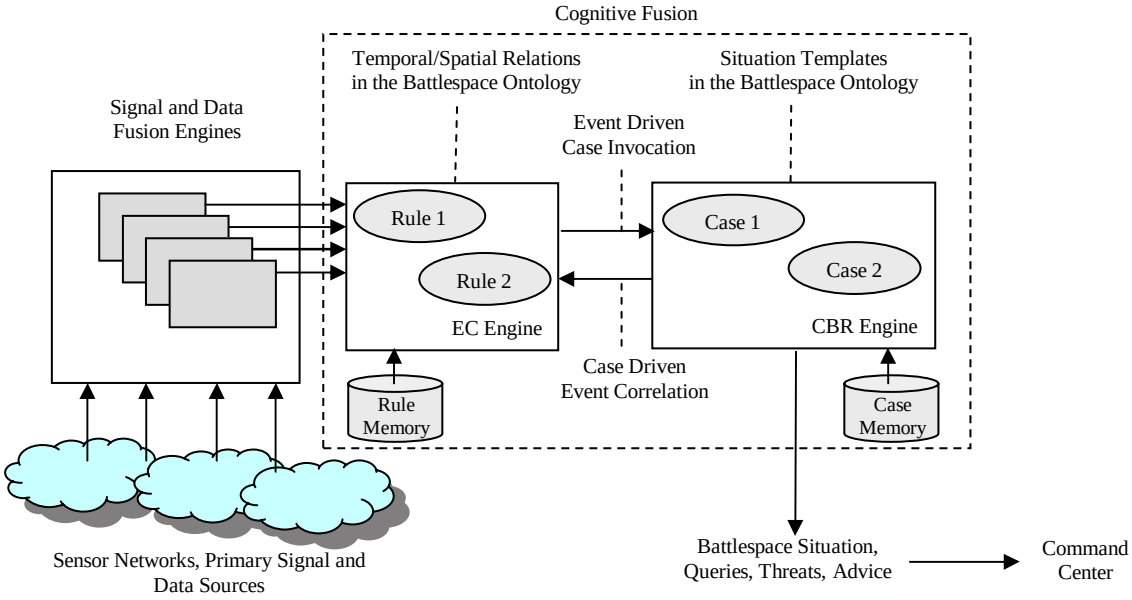

Figure 3. An Integrated EC/CBR Architecture

5. Conclusions and further work

In this paper we discussed an architecture for dynamic event-based situation awareness. The contributions of the paper are (i) introducing dynamic cases, (ii) proposing the EC/CBR integration model, (iii) analyzing the specifics of dynamic situation awareness in comparison with traditional event management in communications networks, and (iv) introducing the multi-level model of dynamic event-based systems.

Further research includes a formal representation of the apparatus suitable for prototype building and testing. Some hard problems are (i) modeling unpredictable situations, (ii) developing algorithms for learning dynamic cases, (iii) developing a situation ontology, and (iv) choosing an appropriate implementation medium. The foundation for the implementation will be a distributed services architecture. The use of standard services and components with well-defined functionality and standard inter-component communication protocols allows the building of open, scalable, and customizable systems. Various technologies could be used for building the infrastructure of the EC/CBR system, e.g. CORBA, RMI, and Jini.

6. References

[1] A. Steinberg, C. Bowman, and F. White. Revisions to the JDL Data Fusion Model. *NATO IRIS Conference Proceedings*, Quebec, Canada, Oct. 1998.

[2] D. Rumelhart. The Architecture of Mind: a Connectionist Approach, in *Mind Readings*, MIT Press, 1998.

[3] J. McCarthy and P. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence, in *Machine Intelligence 4*, American Elsevier, 1969.

[4] F. Pierry and R. Reiter. Some Contributions to the Situation Calculus. *Journal of ACM*, 46(3), 325-364, 1999.

[5] Y. Lesperance, H. J. Levesque, et al. A Situation Calculus Approach to Modeling and Programming Agents, in *Foundations and Theories of Rational Agency*, Kluwer, New York, 1997.

[6] H. J. Levesque, R. Reiter, et al. GOLOG: A Logic Programming Language for Dynamic Domains. *Journal of Programming*, 31, 59-84, 1997.

[7] D, Wen-Yu, X. Ke, L. Meng-Xiang. A Situation Calculus-based Approach To Model Ubiquitous Applications, Cornell University e-print Archive, 2003.

[8] S. Mahoney and K. Laskey. Constructing Situation Specific Networks, in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998

[9] G. Jakobson and M. Weissman. Real-Time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints. *Integrated Network Management IV*, IEEE Press, 1995.

[10] L.Lewis. *Managing Computer Networks: A Case-Based Reasoning Approach*. Artech House, 1995.

[11] *D. McGuinness*. Ontologies and Online Commerce. *IEEE Intelligent Systems*, Vol.16, No 1, 2001.

[12] G. Jakobson, M. Weissman, L. Brenner, C. Lafond, and C. Matheus. GRACE: Building Next Generation Event Correlation Services, *2000 IEEE Network Operations and Management Symposium Proceedings*, April 2000.

[13] J. Kolodner. *Case-Based Reasoning*. M. Kaufman, 1993.

[14] D. Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science 7*, 1983.

[15] G. Dreo and R. Valta. Using Master Tickets as a Storage for Problem-Solving Expertise, in *Integrated Network Management IV*, Elsevier Publishers, 1995.