The Technology and Practice of Integrated Multi-Agent Event Correlation Systems

Gabriel Jakobson
Smart Solutions Consulting
Brookline, MA
617-734-5576
info@smartsolutionsci.com

Abstract—*This paper gives a tutorial overview of the main concepts of Event Correlation and shows how the traditional solutions of Event Correlation can be extended using Agent Technologies. We will show that building hierarchies of inter-operating multi-agent event correlation systems provides flexible, scalable, and reliable architectural solution, which fits the cognitive as well operational requirements of managing complex networks and systems. In addition to Event Correlation and Agent technologies, the paper demonstrates how Ontology is playing a critical role in building knowledge intensive Event Correlation components. The paper also discusses the practical use of Multi-Agent Event Correlation in different network and service management domains.*

1. INTRODUCTION

Event Correlation as a branch of Computer Science and Information Technology has been a focus of extensive research and practical applications over the last 10 years. Conceptually, Event Correlation is a real-time event analysis procedure, which assigns new meaning to the set of discovered events. Practically, it is a software platform that enables real-time management of complex networks and systems, including fault, performance, configuration, and security management. Technologically, it is a set of Information Technology tools, which among others incorporate the methods of Artificial Intelligence, Knowledge Management, Distributed Systems, and Object Orientation.

The impetus for Event Correlation research was motivated by the fact that very often complex systems, being under operational stress, malfunctioning of their internal components, or being target of malicious attacks, produce large number alarms, which analyzed independently without recognizing the synergy between multiple information sources, do not reveal their actual internal situation of the system. In addition, large number 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 noticed too late. Failure to capture the sequences of time-dependent events makes hard.to see the trends in the system internal processes and predict the potential future system behavior. Such objective reality helped to define the following four major utility areas of event correlation:

- Surveillance
  - Context-sensitive event filtering
  - Intelligent information fusion
  - Event generalization and discovery of complex situations
- Fault Diagnostics
  - Fault root cause localization and identification
  - Generation of corrective actions and trouble ticketing
- Security Protection
  - Intrusion detection
  - Fraud prevention
- Performance Monitoring
  - Analysis and time-dependent trending of events
  - Performance and QoS degradation discovery

Event correlation has become a widely accepted technology for managing the complexity of modern telecommunication and enterprise. Beyond network management domain, new applications are emerging in the defense and security areas, including intelligent information fusion, intrusion detection, battlefield management, etc. Several new research directions include multi-agent event correlation architectures, knowledge intensive event correlation based on ontology, hierarchical multi-level cross-correlation, very high-speed correlation engines, inexact correlation, automatic discovery of correlation rules, and others. Although we will highlight major principles and technologies of event correlation, the focus of this tutorial will be on several new challenging research topics, namely, multi-agent event correlation, the role of ontology, and extending the principles of event correlation towards dynamic networks and systems. We will show that building hierarchies of inter-operating multi-agent event correlation systems provides flexible, scalable, and reliable architectural solution, which fits the cognitive as well operational requirements of managing complex networks and systems.

KIMAS 2003, October 1-3, 2003, Boston, MA, USA.

2. Dynamic Systems and Intelligence

There are two fundamental concepts that are in our focus of attention through this paper – Dynamics and Intelligence. We will show that there is a strong, and not accidental synergy between some types of systems that are called Dynamic Systems and systems that are called Intelligent Systems. We will define the dynamic intelligent systems, describe their major features, and show how real-time event correlation systems fall under the category of dynamic intelligent systems. We will examine more specific concepts, such as intelligent agents, distributed systems, knowledge intensity, ontology, and show how all these concepts form a framework for Integrated Multi-Agent Event Correlation Systems.

In very general terms (and for the sake of the topic of this paper), the dynamic systems are considered as systems that:

- Change their internal organization, like parameters, configuration, states, etc. in time
- They consume and/or redistribute resources, and
- They Interact with the rest of the world

In addition to the above-mentioned qualifications, some specific dynamic systems, namely the living (biological) systems have a lifecycle of their existence and the reproduction capability.

There is no good single-sentence definition for the Intelligent Systems, although, the following five features are usually contributed to them:

- The Intelligent Systems are capable reflecting (interpreting) and modeling the World, including themselves
- They are organizing and planning their behavior for survival and achieving goals
- They learn, discover things, and improve their skills
- The intelligent systems are able to communicate with others, form unions and federations, and cooperate under different organizational paradigms
- And finally, they are able to explain their behavior and the results of their actions.

We intentionally left out some complex abstract features of human mind like intuition, intention, insights, feelings, understanding of beauty, etc., which are out of scope of this tutorial.

Although synergy between dynamic systems and intelligence has deep and complex nature, we will focus on two mutually affective ties: on one hand, we see system dynamics as pre-requisite for intelligence, and on other hand, intelligence as a factor defining a purposeful behavior of dynamic systems. Assuming this synergy, we will introduce In the Section 3, two models, the Event Model and the Correlation Model as models describing the dynamic and intelligent features of the Multi-Agent Correlation framework for managing dynamic systems.

Let's consider in more detail the dynamic features of complex systems. Until recently, many aspects of them were handled as static ones. The most common example is the network configuration, except, probably infrequent network topology updates. In reality, many features of complex systems should be considered as dynamic ones, such as

- Dynamic system topology re-configuration
- Changes in the functionality of the nodes, e.g. changes defined by mobile agents
- Transformations of link semantics
- Dynamic adjustment of behavioral goals and agreements
- On-fly selection of system optimization criteria
- Dynamic re-specification of system interfaces

In the telecommunication network management domain examples of dynamic networks include

- Dynamically re-configurable networks
- Active (programmable) networks
- Dynamic VPN (virtual private networks)
- Mobile and survivable defense networks
- Reconfigurable cellular networks (e.g. dynamic channel allocation)

Monitoring in real-time of services, service level agreements (SLA), resources, and quality of service is becoming a critical aspect of successful service provisioning. Any fault or degradation of the network may result in violation of the SLAs or even halt the requested service. Dynamic aspects of service management may include:

- On-the-fly changes in service definitions
- Dynamic re-specification of SLAs
- Changes in resources
- Requests for rapid near real-time deployment of new services

3. The Basics of Event Correlation

All major players in the network management arena either have developed their own, usually embedded, event correlation procedures or have used event correlation products such as InCharge [1], NerveCenter [2], ILOG [3], ART-Enterprise [4], NetExpert [5], and NetCool [6]. Various approaches to event correlation exist, including rule-, case-, and model-based reasoning, finite-state machines, petri nets, and binary coding methods. Several general issues for future directions in event correlations, including distributed event correlation and global correlation, have been discussed in [7].

We will follow the event correlation model described in [7], 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. The event itself is 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 (base) events and the derived (correlated) events. The raw events are external events originated outside the correlation process, while the derived events are results of a correlation process. Depending on the nature of the operations performed on events, different types of event correlation could be defined, including event compression, filtering, suppression, generalization, specialization, temporal relations, and event clustering. The overall event correlation process is run by the Correlation Engine, which uses Network Configuration Information, Correlation Knowledge, and special support functions, like event counting.

Each event correlation process has an assigned correlation time window, a maximum time interval during which the component events should happen. The correlation process will be 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. Generally, time is a critical factor in the correlation process in several different accounts. First, very often the security 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 attack events. In addition, the physical latencies in the communication lines could distort the actual order of incoming events causing incorrect pattern matching.

The adopted approach to event correlation uses the principles of model-based reasoning originally proposed for troubleshooting electronic circuit boards. The idea of the model-based approach is to reason about the system based on its structure and functional behavior. The structural representation captures the specifications of the components that the system is built upon and the relations between the components, such as class, containment, and connectivity relations. The behavioral representation describes the dynamic processes of event propagation and correlation. These processes are described using correlation rules. Each rule activates a new event, which in its turn may be used in the firing condition of the next correlation rule.

Let's consider an example of a correlation rule. In a Mobile Switching Center (Figure 1) Physical Port Y of the Digital Cross-Connect System DCS B reports a carrier-group-alarm (CGA) "yellow", while the Physical Port X of the DCS A in the Central Office reports CGA "red". It is also known that there is a T1 trunk between ports X and Y. The corresponding correlation rule might look as follows:

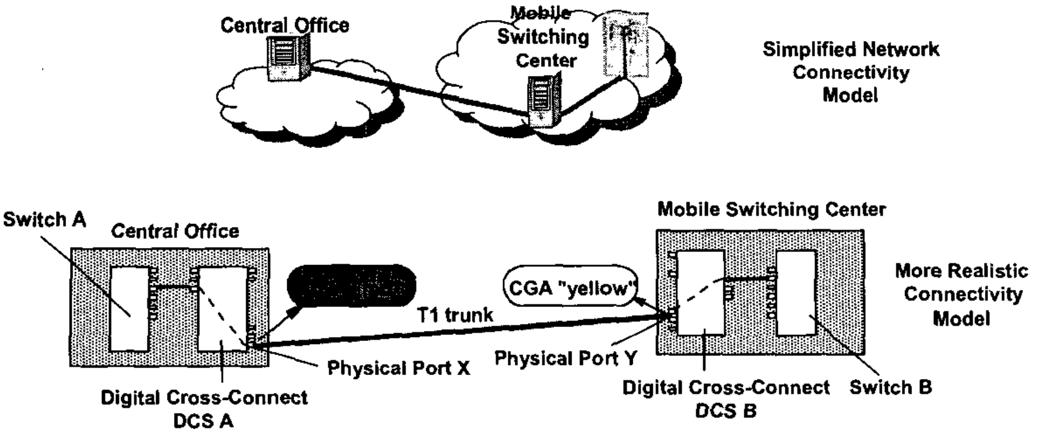

Figure 1 – Facility Disconnect Correlation Example

IF There is a CGA Red Alarm from the Physical Port X belonging to the DCS A
AND There is a CGA Yellow Alarm from the Physical Port Y belonging to the DCS B
AND The ports X and Y are connected by a T1 trunk

AND The alarms CGA RED and CGA Yellow are coming within 3 seconds
THEN There is facility disconnect between switches A and B

4. ONTOLOGY

Ontology is a methodology (and ultimately, a formal language) of systematic conceptualization and classification of objects, object relationships and actions over the objects in a specific Domain. Taking its roots on classification of things from natural sciences, and on description of semantic classes and object orientation from Artificial Intelligence, Ontology is becoming a powerful formal tool of organizing knowledge. In principle, any entities, procedural or declarative, static or dynamic, abstract or real, could be a subject for ontology-based classification.

For example, in the network management domain ontology is used for describing class hierarchies of network elements (NE), such as switches, digital cross-connect systems, channel service units, trunks, routers, bridges, etc. In the service management domain ontology could be built over service elements (SE) such as service types, customers, service level agreements, bills, collection procedures, etc.

It is important in ontology to distinguish between so-called upper level ontology [8], which describes abstract classes of domain entities, and instances of abstract classes, which represent particular "real" element. Following the inheritance paths in the class hierarchy of the ontology, the constraints, attribute values, and attached procedures of a class (parent) will be passed to its subclasses (children). In addition to the class-subclass relations, which are, essentially used to build element classification hierarchies in ontology, many other physical or abstract relations could be used between elements. For example, network element connectivity relation could be used for building network configuration models, while relation "provided-by" could be used for building service provisioning models.

5. AGENTS AND MULTI-AGENT SYSTEMS

Agents and multi-agent systems have found its way into the mainstream of artificial intelligence research [9}. An agent is considered as a virtual (software) or physical entity that can perceive its environment, plan its goal-directed behavior, act to achieve the goals and communicate with others in an autonomous way. It possesses the required knowledge and skills, and is capable of learning. As one can notice, this agent definition fits the general requirement of intelligent systems discussed in the Section 2, except, with one additional feature, namely autonomous behavior. This feature of autonomy coupled with the capabilities of inter-agent communication serves as a basis for construction of multi-agent systems, which ultimately has lead to the notion of Distributed Artificial Intelligent (DAI) Systems. Since the introduction of DAI in 80's, many different research activities has focused on cooperative, mobile, reactive, information retrieval, and other types of agents. The fundamental features of agents - perception, planning, collaboration, autonomy, and learning have always been in the focus of research on agents. Despite of the heterogeneity of different agent architectures, we will introduce a generic agent architecture (Figure 2), which serves as a basis for instantiation of different domain specific agents. The architecture contains core components: Planning Engine, Action Engine, Explanation Engine, Source Perception Modules, and Functional Modules to perform basic agent goal-directed tasks mentioned above. In order to support

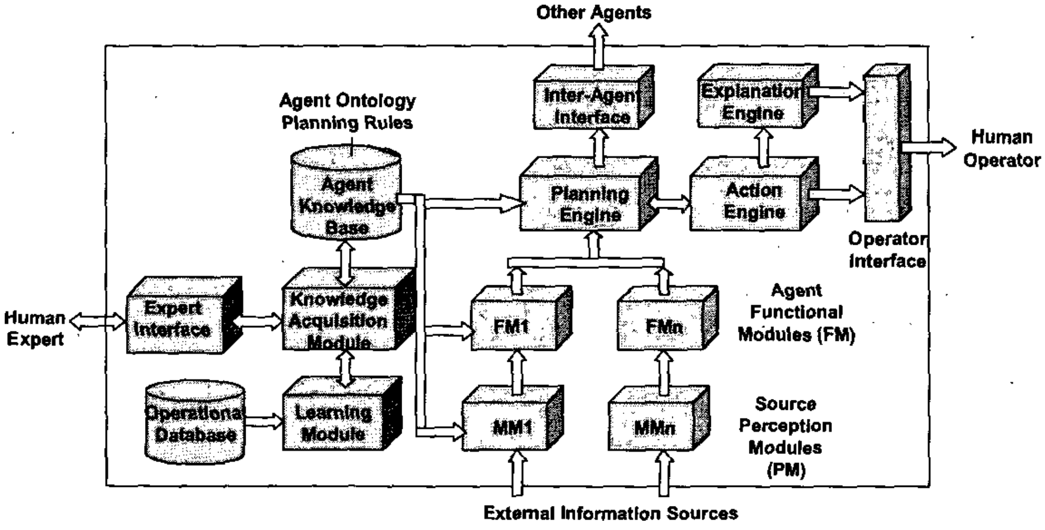

Figure 2 – Agent Architecture

execution of these tasks, the architecture includes several knowledge acquisition and learning components. Agent communication with other agents, external information sources, and human experts and operators is mediated via dedicated interfaces.

6. Distributed Multi-Agent Event Correlation

One of the most fundamental changes in the architecture of enterprise network management systems is the move from embedded, monolithic, and loosely coupled architectures toward distributed, open, component-based architectures. The use of standard system services (components) with well-defined functionality and standard inter-component communication protocols allows the building of open, scalable, and customizable systems. The encapsulation of the idiosyncrasies of components and the easy addition, replication, and replacement of components provide an effective environment for developing scalable, fault-tolerant, and high-performance systems. Various technologies can be used for building the infrastructure of distributed management architectures, including CORBA, .NET, DCOM and Java RMI.

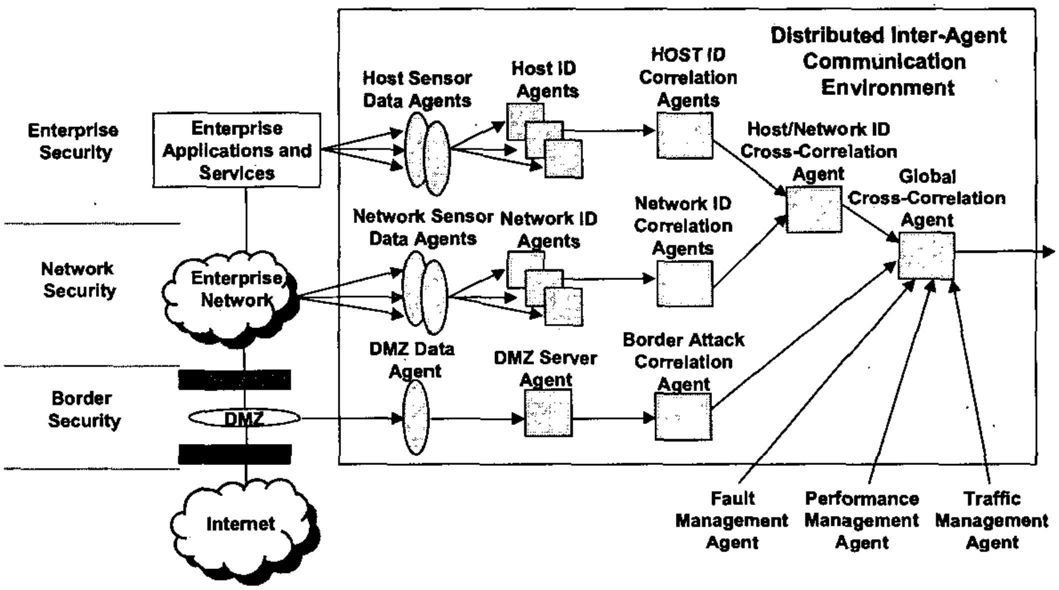
Distributed Inter-Agent Communication Environment
Enterprise Security
Enterprise Applications and Services
Host Sensor Data Agents
Host ID Agents
HOST ID Correlation Agents
Host/Network ID Cross-Correlation Agent
Global Cross-Correlation Agent
Network Security
Enterprise Network
Network Sensor Data Agents
Network ID Agents
Network ID Correlation Agents
Border Security
DMZ
DMZ Data Agent
DMZ Server Agent
Border Attack Correlation Agent
Internet
Fault Management Agent
Performance Management Agent
Traffic Management Agent

Figure 3 - Enterprise Intrusion Detection Based on Multi-Agent Event Correlation

We illustrate the architecture of a distributed multi-agent based event correlation system with an enterprise intrusion detection application (Figure 3). Figure 3 identifies three security levels: Enterprise Border (DMZ) security, Enterprise Network Security, and Enterprise Applications and Services (Host) security layers. On each layer security data is collected by corresponding Data Collection Agents, then analyzed by Intrusion Detection (ID) Agents, and finally correlated by Event Correlation Agents. Figure 3 also depicts two higher correlation levels to perform the tasks of cross-correlation between host and network ID events, and between the intrusion detection alarms and network fault, performance and traffic alarms. As we, see a quite complex dynamic system management structure could be built using a multi-agent distributed architecture. Key for this architecture is provisioning a fast inter-agent communication environment. For example, such environment could be built using CROBA Event Notification Service. The Notification Service enables sophisticated event passing interfaces between agents. The interfaces are mediated via Event Channels that define several important event management functions, including asynchrony, event subscription, multicast event routing, event filtering, quality of service, and structured events. The output of one channel can be chained to the input of another channel to create multi-level event notification chains between the agents.

7. Conclusions and Future Work

Integration of agents and multi-agents, ontology-based intelligent information processing, and event correlation technology into one integrated framework supported by an open and scalable distributed environment provides efficient tools for managing complex dynamic systems. In the future developments, three major forces will drive multi-agent based event correlation research and application: event correlation: (a) globalization of the utility of event correlation, (b) distribution of the event correlation service architecture, and (c) the use of the Internet. Globalization of event correlation breaks the existing application and technology boarders of event correlation, and primarily it means applications in the homeland security and defense areas, such as intelligent information fusion, battlefield management, and infrastructure protection.

While various approaches to event correlation exist, we see the need for important technological advancements of event correlation in the following areas:

- Explanation of the content of the derived solutions and their logical reasons.
- Discovery of correlation knowledge, e.g. learning correlation rules.
- Extension of correlation paradigms with hypothetical reasoning, inexact (fuzzy) knowledge, and the logic of time, space and action

.

Different solutions can be used to implement the same network management functionality, based on alternative reasoning paradigms. A paradigm will be selected based on the specificity of the tasks, the operational context, and the goals of the management process.

REFERENCES

[1] S. Yemini, S. Kliger, Y. Yemini,D. Ohsie, "High Speed and Robust Event Correlation," IEEE Communication Magazine, May 1996.

[2] VERITAS NerveCenter 3.5, http://www.veritas.com/products/nervectr/.

[3] ILOG Rules, http://www.ilog.com/products/rules/whitepaper.pdf.

[4] ART*Enterprise™, http://www.brightware.com.

[5] NetExpert, http://www.osi.com/.

[6] NetCool, http://www.micromuse.com/index.html.

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

[8] N. Guarino, "Formal Ontology and Information Systems," Proceedings of the First International Conference on Formal Ontology in Information Systems, 6-8 June 6-8, 1998.

[9] Jacques Ferber, Multi-Agent System: *An Introduction to Distributed Artificial Intelligence,* Harlow: Addison Wesley Longman, 1999.

[10] G. Jakobson, "Advanced Multi-Layered Intrusion Detection for Enterprise Networks Based on Distributed Event Correlation," *IEEE Fall 2002 Homeland Security Conference Proceedings*, October 2002.