Correlation DialTone—Building Internet-Based Distributed Event Correlation Services

Gabriel Jakobson and Girish Pathak
GTE Laboratories Incorporated
40 Sylvan Road, Waltham, MA 02541-1128, USA
{gjakobson, gpathak}@gte.com

Abstract. In this paper, we propose an Internet-based information correlation service called *Correlation DialTone* that is designed to increase the utility of event-based information. This service concept opens new business opportunities for Internet-based information services, such as Stock Market Correlation DialTone, Health Care Correlation DialTone, and Home Security Correlation DialTone. We present the component-based architecture of the distributed event correlation service and its component services: correlation subscription, data mediation, event parsing, event correlation, event delivery, and event notification services. We describe process/knowledge representation models and implementation of the distributed correlation service using CORBA, XML, Java, and model-based reasoning technologies. In addition to discussing the concept, the proposed service architecture, and the candidate applications, we also describe an implementation of a distributed event correlation system supporting the Correlation DialTone service.

1 Introduction

The Internet is becoming a universal information transport and service medium. It will soon be possible to connect any business, home, device, process, transportation vehicle, living body, or any other object (in any pragmatic information-producing sense) to any information consumer, human or machine.

While the amount of information flowing over the Internet is enormous and increasing exponentially, the utility of the information is growing much more slowly. This situation presents two related problems: “information flood” and low information utility. Simply stated, how does one effectively process the huge volume of information and extract, fuse, and interpret information in a way that is truly useful? In this paper, we propose partial solutions to these problems that focus on *events* and their *correlation.*

Depending on the way information is generally handled, one can see an evolutionary path in the use of the Internet:

- Information transportation
- Transaction processing
- Dynamic information change management

C. Linnhoff-Popien and H.-G. Hegering (Eds.): USM 2000, LNCS 1890, pp. 93-104, 2000.

The majority of Internet applications today provide basic information transportation utility. Good examples include sending e-mail, transferring files, retrieving HTML documents, etc. New opportunities for utilization of the Internet are related to conducting business, such as banking, trading, commerce, and other transactions. In the future, we will see rapid expansion of the Internet for *dynamic information change management – event management*. The dynamic information change includes events such as notifications of physical processes, system status changes, fault alarms, changes in sensor measurement data, changing news items, surveillance information changes, etc. The Internet will become the predominate medium for generating, notifying, transporting, analyzing, correlating, and presenting all forms of dynamic information change. Important applications in dynamic information change management include automatic process control, network fault and performance management (system, device, and home), security management, location management, workforce mobility management, etc. All of these require collection and correlation of events.

In this paper, we propose an Internet-based information correlation service that we call Correlation DialTone (CDT). The CDT service, based on real-time, distributed event correlation technology, is designed to increase the utility of event-based information and is expected to have applicability to several new business opportunities, such as Stock Market CDT, Personal News CDT, and Home Security CDT. We will present the concept of CDT, the distributed service architecture of CDT, and the underlying real-time event correlation technology that will be utilized to correlate Internet events; and, finally, we will describe the components of a distributed event correlation system developed at GTE Laboratories.

As applied to Internet-based services, ideas similar to CDT have been expressed in the Keryx system from HP [1].

Event correlation technology that will be used for CDT is a widely accepted solution for managing the complexity of modern telecommunication and data networks. It has been used for various network management tasks, but the majority of its applications have been for network fault management. Specific correlation applications have been developed to manage switched, SS7, wireless, ATM, SONET, IP, and other networks. 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 [2], NerveCenter [3], ILOG [4], ART-Enterprise [5], NetExpert [6], and NetCool [7]. 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 [8].

2 Correlation DialTone—Concept and Applications

Real-time event correlation has been used for well over a decade with applications in various fields, not the least of which is network management. The Internet is creating a plethora of new opportunities for event correlation while also presenting new challenges. This evolving landscape has given rise to the Correlation DialTone concept. Correlation DialTone is an Internet-based event correlation service that allows real-time analysis of various information sources and managed systems/devices connected to the Internet (Figure 1). At its core it is an intelligent, knowledge-based information processing service, capable of recognizing correlations between events using application domain knowledge and data about the internal structure (topology) of the managed sources and objects. Subscribers to CDT will be able to identify the information sources of interest, specify event filtering and correlation conditions, select how they want to be notified about the correlation discoveries, and determine the information notification presentation media and form

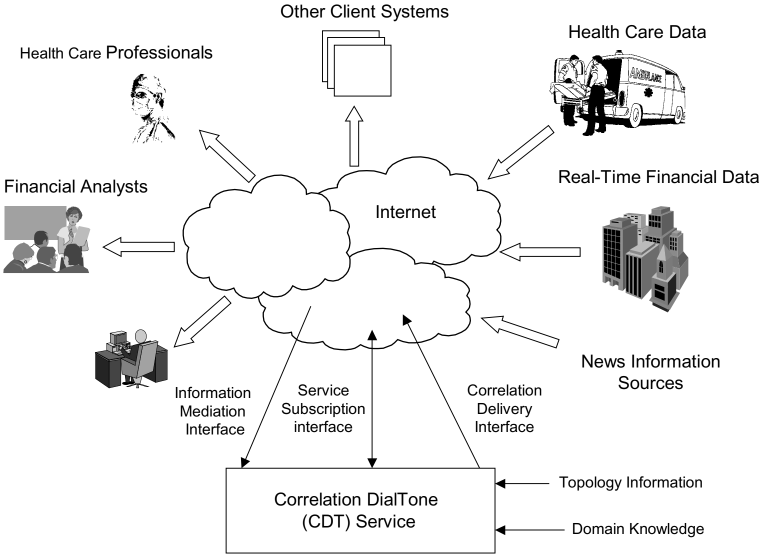
Other Client Systems
Health Care Professionals
Health Care Data
Financial Analysts
Internet
Real-Time Financial Data
News Information Sources
Information Mediation Interface
Service Subscription interface
Correlation Delivery Interface
Correlation DialTone (CDT) Service
Topology Information
Domain Knowledge

Figure 1. Correlation DialTone Conceptual Architecture

CDT is a distributed system with different component services for correlation subscription, event mediation, parsing, correlation, correlation delivery, system administration, and other component services. The correlation subscription service allows the clients to define their requests for CDT, such as the information sources, the managed objects, the type of correlation, the way the client will be notified, etc. The event mediation service provides connectivity to information sources and managed objects. It also performs the tasks of communication protocol and data adaptation. The event correlation service performs real-time event correlation. This service will be described in more detail in the following sections. The message parsing service performs event analysis functions that may include steps of different complexity of syntactic and semantic processing, depending on the nature of the event sources to be correlated. The event delivery service returns the CDT results to the client, while the administration service performs such tasks as billing, security, and the CDT internal management tasks.

One example of CDT applications is the Stock Market CDT—a system for real-time stock portfolio margin management. The system gets real-time feeds of predefined stock quotes and correlates them to determine adjusted and federal margin calls. The system will be able to perform single stock, diversified portfolio, or full portfolio volatility calculations, and may contain an expert system advisory component to provide buy/sell recommendations.

Other examples of CDT applications include:

- Personalized News CDT—customizable, real-time filtering and correlation of structured news data.
- Home Security CDT—correlation of data from home fire and intrusion detection, surveillance, and other devices.
- Medical Data CDT—monitoring and correlation analysis of bodily statistics, collected and relayed in real-time back to a medical center.

The heart of the CDT is the event correlation process. The event correlation process is defined as a conceptual interpretation procedure that assigns new meaning to a set of events [9]. It may range in complexity from simple event compression to complex pattern matching across asynchronous events. Algorithmically, event correlation is a dynamic pattern-matching process over a stream of events. In addition to the real-time events, the correlation patterns may include the topology of the information source (e.g. network connectivity), diagnostic test data, data from external databases, and other ancillary information. Event correlation enables several event management tasks:

- Increasing the semantic content of information through generalization of events.
- Fusion of information from multiple sources.
- Real-time fault detection, causal fault diagnosis, and suggestion of corrective actions.
- Ramification analysis of events and prediction of system behavior.
- Long-term trending of historic events.

3 Distributed CDT System Architecture

One of the most fundamental changes in the architecture of telecommunication and enterprise network management systems is the move from embedded, monolithic, and loosely coupled architectures toward distributed, open, component-based architectures. The use of standard services (components) with well-defined functionality and standard intercomponent 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 multiparadigm, faulttolerant, and high-performance systems. Various technologies can be used for building the infrastructure of distributed network management systems, including CORBA [10], DCOM [11], and Java RMI [12]. Figure 2 shows a general overview of the CDT system architecture. The architecture is based on the following principles:

- Encapsulation of implementation idiosyncrasies of the different components (in our case, CORBA objects).
- Use of a standard event presentation (CORBA structured events).
- Use of a common knowledge/data transportation format (XML).

Thanks to these features, one can build customized management systems of different functionality, scale, and complexity. Different instances of the domain level services can be used, as long as they all satisfy overall functional and data semantic constraints. For performance or functional reasons, multiple processes of the same service could be launched. For example, a hierarchy of event correlation processes could be created. This hierarchy could be used to implement a multilevel system management paradigm, e.g., to implement local and global correlation functions.

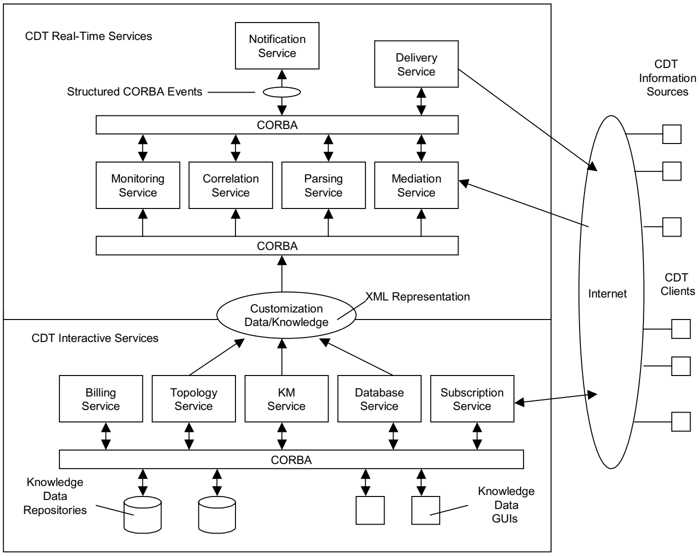

Figure 2. Distributed CDT System Architecture

In this CDT architecture, we divided the services into Real-Time Services (Mediation, Parsing, Correlation, and Event Notification Services) and Interactive Services. The latter ones are grouped into Support (Security, Billing, Subscription) and Customization (Topology, Knowledge Base, Modeling, Database) Services. This division supports our desire to provide fast channels of real-time event processing, and to make available interactive services, on an on-call basis, to provide required knowledge, models, procedures, and data in support of the real-time processes. The real-time services are sensitive to performance, and require a certain level of fault tolerance.

The Mediation Service provides Internet connectivity to CDT information sources. The incoming raw events (messages) are parsed by the Parsing Service. This might involve procedures stretching from analysis of simple structured data or text to complex semantic analysis of (near) natural language expressions, e.g., processing of news items. The Correlation Service performs the functions of real-time event pattern matching; processes event objects, topology, and other data; and executes actions predescribed by the correlation rules.

The Event Notification Service plays a special role in the architecture. It facilitates communication between the real-time components of the architecture. It enables sophisticated event passing interfaces between distributed objects—the producers and consumers of events. The interfaces are mediated via event channels that allow decoupling of producers and consumers in the sense that they possess no knowledge about each other. The CORBA standard for the Notification Service, the OMG's COSNotification Service, defines several important features of the Notification Service, 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 a notification chain as shown in Figure 3. Each of the nodes in a notification chain may cache events, take actions, perform some transformation on the events, and forward them along the chain. Services may select relevant events via filters. It becomes easier to replace these chained services with newer or alternate versions because the interaction is decoupled. It is easy to add supporting functions, such as validation, by creating a service and having it subscribe to a preexisting channel.

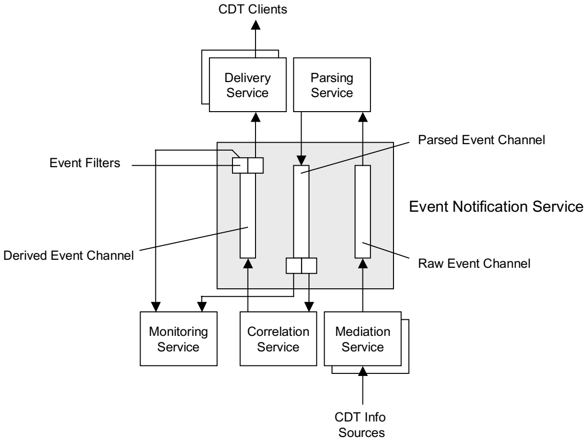

Figure 3. Chaining Services for CDT

4 The Event Correlation Model

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 could stretch 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 manifestation of a change in the state of an object, system, or process. 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 [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 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 component event instantiates a new correlation time window for some correlation.

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 (correlation), which in its turn may be used in the firing condition of the next correlation rule.

On a phenomenological level, a correlation is a statement about the events happening on the network. On a system internal level, a correlation is an object-oriented data structure that contains its component objects and attributes. All correlations are organized into a correlation class hierarchy. The root node of the correlation class hierarchy describes the basic correlation class. The terminal nodes represent correlation classes, which will be instantiated each time particular conditions are met in the stream of incoming events. A correlation rule defines the conditions under which correlations are asserted. Different correlation rules may lead to the assertion of one and the same correlation. The subsequent application of correlation rules, instantiation of correlations, and consumption of produced correlations by the next rule describe the event propagation process.

5 Correlation Service

The Correlation Service (see Figure 4) runs all of the processes required for real-time event correlation. It contains the following subcomponents:

- Correlation Engine—performs the functions of real-time event correlation; processes event objects, topology, and other data; and interprets the state of the network. The Correlation Engine uses CLIPS [15] as the underlying real-time inferencing tool.
- RT (Real-Time) Event Export Module—caches, in real time, the state of the Correlation Engine (including the input events passed to the Correlation Engine and the resulting derived events) and passes it to the Event Database for further use by the Explanation Engine.
- RT Topology Export Module—handles the real-time interface to the Network Topology service using asynchronous CORBA calls.
- Action Service—application-specific services that provide asynchronous scripting (Tcl, Perl, Java), external function calls (e.g., in C), database queries, access to test equipment, and other functions to the Correlation Engine.

By subscribing to the Event Notification Service, the Correlation Engine receives a variety of input events, such as parsed base events from the Message Parsing Service, results from database queries or other external actions, and derived events generated by other correlation processes. All real-time events are presented as CORBA structured events. To proceed with the correlation process, the Correlation Engine gets required correlation knowledge from the Knowledge Management Service. This information is transported in XML. The output of the Correlation Engine is derived structured events pushed to the Event Notification Service. Information in the derived structured events may be used to filter and select items of interest to other components. Specific information in the derived structured events may include:

- Requests to other services to take actions or fetch data.
- Messages that should be added to event lists.
- Messages that should be removed or modified in event lists.
- Input to other correlation processes performing global correlation.
- Status information, which may be logged or ignored.

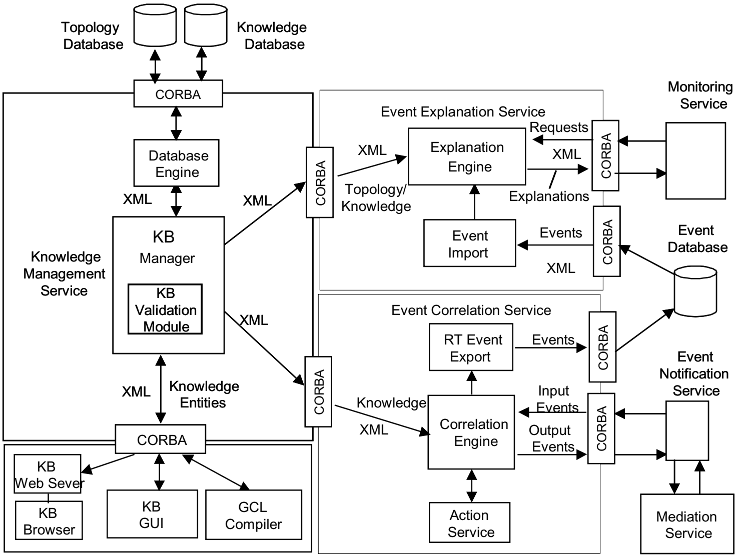

Figure 4. Correlation, Explanation, and Knowledge Management Services of CDT

6 Explanation Service

The Explanation Service (Figure 4) is used to analyze conclusions or situations recognized by the Correlation Engine. Some of the things this component is responsible for are:

- Finding events subsumed by a selected event.
- Finding independent events.
- Performing detailed non-real-time analysis of the cause of an event.
- Displaying a causal tree of a derived event.
- Displaying rules, messages, and other information related to the derived event.
- Displaying advisory text associated with derived events.

The Explanation Service gets requests for explanations from CDT customers via the Monitoring Service. The Explanation Engine has access to the Event Database, and, in the same manner as the Correlation Engine, it is provided with the required knowledge and topology information. As shown in Figure 4, the event explanation and correlation processes are separated and run by different engines. They only share access to the common event database, and to the correlation knowledge and the topology information. This solution better accommodates the real-time nature of the correlation process and the interactive (on-call) service of the Explanation Engine.

7 Knowledge Management Service

The Knowledge Management (KM) Service contains the following subcomponents:

- Knowledge Base (KB) Manager—serves and verifies the Knowledge Base and mediates physical database access.
- KB Browser—allows simple querying and reporting about GRACE and the Knowledge Base via the Web.
- KB GUI—a graphical user interface for editing KB entities, while maintaining consistency and correctness.
- GCL Compiler—provides text-based interface to define and edit KB entities described in the GRACE Correlation Language (GCL).

Database Engine—provides interactive access (read and write) to the knowledge and topology databases.

The Knowledge Base contains a variety of knowledge, including correlation rules, finite-state machines, and specification of application domain entities and their relationships. The standard language used for describing the content of the KB is XML.

XML, the Extensible Markup Language, is becoming the de facto standard for transporting data and knowledge between distributed components. XML represents arbitrary semantics as strings. It is not necessary to predefine the contents of these strings for the sake of the transport medium, making XML ideal for transferring data of arbitrary semantics over CORBA. While the distributed systems will define their framework in CORBA IDL, they will define much of the data semantics in XML. This approach will allow components of the system to be decoupled supporting a consistent knowledge and data transport mechanism. A sample of this transportation schema based on XML as an intermediary data/knowledge representation language and implemented in the CDT system is shown in Figure 5.

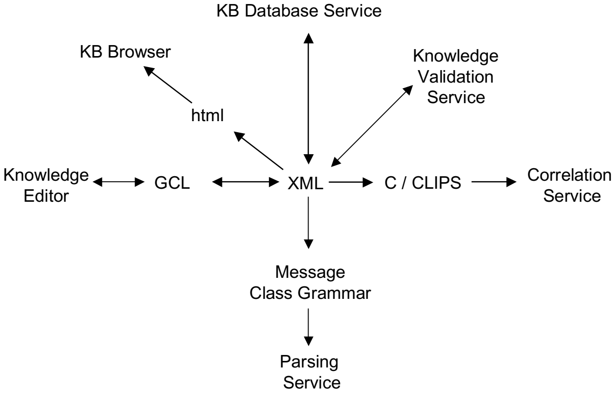

Figure 5. Data/Knowledge Transportation via XML Representation

8 CDT—Implementation

CDT is built from distributed CORBA components that communicate via IIOP (Internet Inter-ORB Protocol). OmniORB [13] is used for ORB programming, while the CORBA COSNotification (with a few small exceptions) was implemented at GTE Laboratories. The rest of the components were programmed in C/C++, Java, XML, and CLIPS. The system runs on Solaris, Linux, and AIX.

As events are passed from the Internet-based information sources or transported from managed systems via the Internet, the incoming stream of events is handled by the Mediation Service. The Mediation Service performs low-level protocol conversions and data adaptation functions, and turns events into a standard CORBA structured event format. Following the event notification chain discussed in Section 3, incoming events are then processed by the Parsing Service. In principle, many different parsing algorithms may be used; in the CDT implementation, however, a declarative knowledge-based method for event parsing was used [14]. At the core of this approach is a declarative message parsing knowledge called Message Class Grammar (MCG). MCG drives a universal parsing engine, allowing it to be customized to the messages of different active elements. The proposed method unifies message processing. It allows easy specification of different message processing tasks such as generalization, classification, specialization, transliteration, and others.

The event correlation model discussed in Section 5 is implemented in the Correlation Service component, which in its core uses an enhanced RETE network-based dynamic rule engine [15].

The Mediation, Parsing, Correlation, Topology, and possible other CDT component services are customized to specific CDT applications by utilization of the customization knowledge base. The content of the customization knowledge base is diverse. For the particular CDT implementation, discussed in this paper, it includes domain element classes, objects, message classes, correlation rules correlations (objects), relations, and finite-state machines. This content of the knowledge base (KB) is developed by CDT domain experts (engineers) using KB development GUIs. The Knowledge Management Component serves and verifies the Knowledge Base and mediates physical database access. It provides the Event Explanation and Event Correlation components with online access to the correlation knowledge and topology. The Knowledge Base editors are used for creating and modifying Knowledge Base entities, while maintaining consistency and correctness of the Knowledge Base.

9 Conclusion

Penetration of the Internet into all segments of the economy, including public and private institutions, the government, and the home, creates real business needs for online information correlation services. In this paper, we introduced the notion of Correlation DialTone—an Internet-based information correlation service. We described the concept, architectural framework, potential applications, and some key technologies for implementing real-time event correlation procedures.

Several research and implementation issues are still open, e.g., efficient processing of unstructured source information (natural language news events), discovery of new correlation rules from large bodies of data (data mining), understanding and incorporation of nontextual information, etc. The market size and the benefits of Internet-based correlation services are hard to overestimate. A deeper study of the full potential of Internet-based correlation services is in the works.

References

[1] Keryx, http://keryxsoft.hpl.hp.com/.HYPERLINK

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

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

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

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

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

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

[8] Jakobson, G., Weissman, M., Brenner, L., Lafond, C., Matheus, C., GRACE: Building Next Generation Event Correlation Services, NOMS 2000, April 2000, Honolulu, Hawaii.HYPERLINK

[9] Jakobson, G., Weissman, M., Real-Time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints, Proceedings of the Fourth IFIP/IEEE International Symposium on Integrated Network Management, May 1995, Santa Barbara, CA.

[10] Siegel, J., CORBA Fundamentals and Programming, John Wiley & Sons, 1996.

[11] Pinnock, J., Professional DCOM Application Development, Wrox Press Inc., 1998.

[12] Downing, T. B., Java RMI: Remote Method Invocation, IDG Books Worldwide, 1998.

[13] OmniORB, http://www.uk.research.att.com/omniORB/omniORB.html

[14] Jakobson, G., Weissman, M., A New Approach to Message Processing in Distributed TMN, Fourth IFIP/IEEE International Workshop on Distributed Systems, October 5–6, 1993, Long Branch, NJ.

[15] Jackson, P., Introduction to Expert Systems, Addison Wesley, 1999.