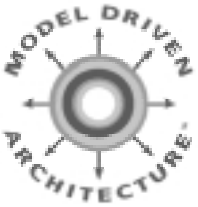

OMG Workshop on Embedded & Real-Time Distributed Object Systems

Workshop Program

MONDAY – January 7, 2002

TRACK I

0900 - 1700 ***CORBA Core and CORBA 3: Architecture and Technical Overview***
Dr. Jon Siegel, Object Management Group

This all-day tutorial covers OMG's Object Management Architecture including CORBA, the CORBAservices and CORBAfacilities, the Domain CORBAfacilities, and an afternoon concentrating on the new specifications included in OMG's latest release, CORBA 3. Starting with a brief look at requirements and needs in distributed computing and how OMG's modeling standards - UML, the MOF, XMI, CWM, and the new Model Driven Architecture (MDA) - fit into the OMG specification suite, the tutorial moves on to cover OMG Interface Definition Language, structure of the Object Request Broker, interoperability and the standard protocols GIOP and IIOP, and interoperability with other middleware platforms. The next section of the tutorial covers the CORBAservices and facilities, and the Domain CORBAfacilities. The afternoon covers the new specifications included in the CORBA 3 release, which fall into three categories: Improved integration with Java and the Internet; Quality of Service Control, and the CORBA Component Model or CCM. The discussion of CCM starts with a closer look at the Portable Object Adapter or POA, on which CCM is based. One attendee at the tutorial will win a copy of Dr Siegel's recent book, CORBA 3 Fundamentals and Programming, published by John Wiley and Sons.

TRACK II

0900 – 1230 ***Introduction to UML: Structural & Use Case Modeling***
Cris Kobryn, Telelogic

This pragmatic tutorial focuses on the 20% of the UML that can be used to specify 80% of common software problems. The first part of this tutorial will cover two kinds of complementary models: use case models and structural models. Use case models specify system behavior as it appears to outside users by partitioning functionality into transactions ('use cases') that are meaningful to users ('actors'). Structural models define the static structure of objects, including their classifiers, relationships, attributes and operations. The tutorial concludes by showing how UML can be customized for platforms and application domains by using profiles.

1330 – 1700 ***Behavioral Modeling*** - Conrad Bock, Kabira Technologies

The second part of this tutorial shows how behavioral models complement structural models by specifying the dynamic behavior of objects, including their collaborations, state histories, and activity models. The tutorial concludes by showing how large UML models can be maintained using model management constructs such as packages, models and subsystems. The latest version of the UML specification, UML 1.4, will be used for all examples.

1030 - 1045 Morning Refreshments

1230 - 1330 Lunch

1500 - 1515 Afternoon Refreshments

TUESDAY, January 8, 2002

TRACK I

0900 - 1230 ***Fault Tolerant Systems Tutorial***

Joey Garon, Vertel Corporation

Priya Narasimhan, Carnegie Mellon University

With CORBA's new Fault Tolerance standard, you can design systems that recover automatically from hardware failures, let users "hot swap" cards in a live system, and provide true 24x7 availability—features especially important in modern, high-performance, high-availability telecommunication networks. The standard defines these features in a way that makes them matters of configuration, so redundancy and recovery occur transparently to application code. All this is achieved following industry-standard protocols that provide interoperability in multi-vendor networks. Fault-Tolerant CORBA defines standards for: redundant servers (replicas); automatic checkpointing; fault detection, notification, and analysis; and recovery from faults.

The tutorial will focus on the following topics:

1. Designing a fault-tolerant system: where do you start?
2. Key issues in object replication, such as replica consistency, fault detection, logging and recovery.
3. The new Fault Tolerant CORBA standard: Interfaces of the new standard, Requirements placed on the implementation, Properties of fault-tolerant objects, Non-fault-tolerant clients of fault-tolerant servers
4. Writing "clean" applications that can be made fault -tolerant easily
5. Survey of the current practices and solutions for fault-tolerance, and their advantages and disadvantages

1330 - 1730 ***Real-Time CORBA Tutorial***

Shahzad Aslam-Mir, Vertel Corporation

CORBA provides a standards-based infrastructure for building flexible architectures. Real-Time CORBA 1.0 extends this infrastructure to provide timeliness properties for fixed priority systems -- typically for "closed," static, control applications. The recent addition of Real-Time CORBA 2.0 provides significant extensions that allow application engineers to "plug-in" alternative workload management and scheduling capabilities -- typically for "open," dynamic, control applications. It also provides a new programming model that is well suited for building control-flow applications having end-to-end timeliness properties.

This tutorial will review:

- Real-Time, including real-time concepts, scheduling examples and timeliness in distributed systems
- Real-Time CORBA 1.0, including definitions, distributing a real-time system and end-to-end predictability
- Real-Time CORBA 2.0: Dynamic Scheduling, including new concepts, scheduling, and programming model

TRACK II

0900 – 1730 ***UML Profile for Schedulability, Performance and Time Tutorial***

Bran Selic, Rational Software

Ben Watson, Tri-Pacific Software

The Unified Modeling Language (UML) is generally recognized as being a very useful tool for modeling the functional characteristics (the "what it does") of a system. However, real-time requirements placed on systems are non-functional in nature. Response time, availability, throughput and bandwidth address characteristics that cannot be expressed functionally. These requirements are more in the realm of time, stability and performance ("when it does it" and "how well it does it"). Another way of saying it is that these are Quality of Service (QoS) characteristics. Historically, UML has ignored non-functional system requirements. If UML is to evolve into a tool for Model Driven Architecture (MDA), it needs some mechanism to address QoS. The UML Profile for Schedulability, Performance and Time is a recent Object Management Group (OMG) standard that provides a set of stereotypes and tagged values to express a system's non-functional, time related characteristics. The profile identifies a way of thinking about QoS and introduces the concept of model processing for various model analysis tools. This tutorial explains the profile and the concepts employed to derive the stereotypes and tagged values. Its concepts include resources and resource usage, time and timing mechanisms, concurrency, performance, and schedulability. Of particular interest for this workshop is a schedulability model for Real-Time CORBA 1.0 and Real-Time CORBA 2.0. The tutorial includes examples for embedded and real-time CORBA applications.

1030 - 1045 Morning Refreshments

1200 - 1800 ***Demonstration Area Open***

1230 - 1330 Lunch

1500 - 1530 Afternoon Refreshments

WEDNESDAY, January 9, 2002

TRACK I

0900 - 1200 ***Notification Service Tutorial***
Bill Beckwith, Objective Interface Systems

The OMG Event Service supports asynchronous exchange of event messages between clients. The OMG Notification Service extends the existing OMG Event Service by adding the following capabilities:

- The ability to transmit events via either a well-defined data structure, Anys, or Typed-events.
- The ability for clients to define filters that specify exactly what events the client is interested in receiving.
- The ability for an event supplier to discover existing and new event requirements from event consumers. This avoids the transmission of events with no consumer interest.
- The ability for an event consumer to discover the types of events offered by suppliers. Consumers can subscribe to new event types as they become available.
- The ability to configure various quality of service properties on a per-channel, per-proxy, or per-event basis.
- An optional event type repository which, if present, facilitates the formation of filter constraints by end-users, by making information readily available about the structure of events that will flow through the channel.

This tutorial will review:

- the relationship of the Notification Service to other OMG specifications,
- the basic capabilities of the Notification Service, and
- several use cases with sample code.

In addition, the ongoing effort to produce a Real-Time Notification Service specification will be discussed.

TRACK II

0900 – 1200 ***System Design: Architectures and Archetypes***
Stephen Mellor, Project Technology, Inc.

The challenges of developing high performance, high reliability, and high quality software systems are too much for ad hoc and informal engineering techniques that might have worked in the past on less demanding systems. New techniques for managing these growing complexities are required to meet today's time-to-market, productivity and quality demands. This tutorial shows you how to:

· engineer the system-wide design to meet performance constraints;
· identify the characteristics of the problem that determine the system design;
· model the system-wide design—the software architecture;
· build archetypes that generate efficient code.

We propose an executable UML as a platform-independent language to build a platform-independent model (PIM) of the application, the details of which, including the logic, are captured in a metamodel repository. The key concept is the software architecture—an abstract machine that executes an executable UML model. The selected software architecture acts as the platform. Weaving the PIM and the platform together yield a platform-specific model (PSM) that may or may not have the same structure as the original object-oriented model. We demonstrate a mapping from an executable UML model into a monitor and control architecture as an example. The mapping is carried out—completely, all the way to the code—by a set of archetypes that traverse the application as stored in the metamodel to produce production code. The result is the capability to generate whole systems from a PIM that has the overall performance properties of the software architecture.

1000 - 2000 ***Demonstration Area Open***

1030 - 1100 Morning Refreshments

1200 - 1300 Lunch

1300 - 1315 ***Opening Remarks*** – Richard Soley, Object Management Group

1315 – 1345 ***Sponsor Presentation*** – IONA Technologies

1345 - 1445 ***Smart Transducers***
Chair: Richard Soley, Object Management Group

Embedded sensor devices are multiplying rapidly in the construction of aircraft, automobiles, mobile consumer electronics, clinical healthcare devices and a host of other systems. Previous "smart transducer" standards (like LIN & CAN) have proven too limited for many of these tasks. Thus OMG is completing standardization of interfaces for embedded smart transducer device standards. This session will explain the problems and constraints forming that standard, and how developers can make use of the upcoming standard in their own systems designs.

Smart Transducers Interface

Thomas Losert, Vienna University of Technology, Real-Time Systems Group

A smart transducer (ST) comprises a hardware or software device consisting of a small, compact unit containing a sensor or actuator element (possibly both), a micro-controller, a communication controller and the associated software for signal conditioning, calibration, diagnostics, and communication). This generic interface for smart transducers allows interoperability between several different embedded systems of different manufacturers while still satisfying the demands on safety and temporal predictability by using the time triggered architecture (TTA). Access to an ST-network takes place through a CORBA-object acting as gateway thus allowing e.g. (re)configuration of running nodes over the internet.

Applying CORBA to Embedded Time-triggered Real-time Systems

Hermann Kopetz., Technische Universität Wien and Shahzad Aslam-Mir, Vertel Corporation

This presentation will highlight a new approach that uses CORBA to span the real-time time-triggered domain. Its key elements that will be described are a service interface, a diagnostic and maintenance interface, and the configuration and planning interface in a distributed control system. The aim of this approach is the simplification of I/O programming and thus the reduction of the software costs of distributed control systems development for automotive, aeronautical, and astronautical systems.

1445 - 1515 Afternoon Refreshments

1515 - 1615 ***Tradeoffs Between Embedded and Enterprise ORBs***
Chair: Bill Beckwith, Objective Interface Systems

Lessons Learned in Implementing a Light-weight ORB

Bill Lloyd, IONA

This talk presents a brief overview of the features a light-weight and very fast ORB suitable for embedded systems. During the implementation, considerable experience was gained with the various implementation trade-offs that must be made in order to achieve both high performance and a small memory footprint. The talk presents some of those trade-offs, the reasons for choosing a particular alternative, and discusses the implementation techniques chosen.

Minimum CORBA Tradeoffs

Ken Black, Highlander Engineering

The minimumCORBA profile improves the suitability of CORBA for many resource-constrained applications by reducing an ORB's static memory footprint. This is principally achieved by removing dynamic capabilities, including the Dynamic Invocation Interface (DII), Dynamic Servant Interface (DSI), and some Portable Object Adapter (POA) policies. Because these removed capabilities can be used to improve scalability, certain applications built to take advantage of full CORBA may actually require less total memory than the same application built with minimumCORBA functionality. Furthermore, some applications may be easier to implement using full CORBA. This presentation will review criteria that should be considered when determining whether minimumCORBA or full CORBA is the best fit for an application, and also discuss architectures in which both minimumCORBA and full CORBA can play a role.

1615 - 1815 *Real-Time Programming and Componentry Models*

Chair: Dock Allen, MITRE

Traditionally, real-time systems have been programmed using approaches such as fixed priority scheduling (as supported by Real-time CORBA 1.0), various types of dead-line oriented scheduling (as supported by Real-time CORBA 2.0), cyclic scheduling, and resource management. However, several new innovative approaches are emerging for programming these applications, and for supporting end-to-end timeliness in new domains. This session will present four of these approaches, their applicability and their real-time characteristics.

Overload Protection For Real-Time CORBA Systems

Niklas Widell, Lund Institute of Technology

Performance-related research regarding distribution aspects of real-time CORBA has focused on object distribution and load balancing. Support for these mechanisms exist in several different implementations. However, they are adequate only when the workload caused by arriving tasks is well below system capacity. It is therefore important to develop overload protection mechanisms so that the system load is maintained at an acceptable level. Our work focuses on the introduction of overload protection in real-time CORBA systems. The main objective is to preserve QoS during short periods of overload. The presentation covers the causes of overload, load monitoring and overload protection measures.

Dynamic closed-loop QoS enforcement - new frontiers in real-time CORBA.

Henry Ni., Anurag Verma, and Shahzad Aslam-Mir, Vertel Corporation

A new trend observed amongst real-time CORBA users is to start to craft closed-loop middleware solutions based on feeding back the observed outputs to self-tune and self-regulate the application so as to maintain the 'contracted' levels of QoS. These mark the emergence of new classes of middleware that is not only QoS aware but also QoS regulating, through the medium of self-tuning. This type of middleware acts both as sensor medium as well as actuator and is possibly the only form of solution that can cater to the next wave of grand challenges posed by new technologies using real-time CORBA. This paper will highlight such research avenues as based on extensive industry requirements.

Managing Peer-to-Peer Applications

Vana Kalogeraki, Hewlett-Packard Laboratories

As computers become more pervasive and homes become better connected, a new generation of applications will be deployed over the Internet. In this model, peer-to-peer applications become very attractive because they improve scalability and enhance performance by enabling direct and real-time communication among the peers. We propose a decentralized management system that manages the peer-to-peer applications and the system resources in an integrated way; monitors the behavior of the peer-to-peer applications transparently and obtains accurate resource projections, manages the connections between the peers and distributes the objects in response to the user requests and changing processing and networking conditions.

Experiences with Middleware for a Networked Embedded Software Technology Open Experimental Platform

Venkita Subramonian and Christopher Gill, Department of Computer Science, Washington University

Networked Embedded Software Technology (NEST) defines a class of Distributed Real-time Embedded (DRE) systems in which there are around 100 - 100000 highly resource constrained computing nodes. NEST-like systems are being used increasingly in fields like advanced avionics and space systems. This calls for collocation and embedding of processing and control software within these computing nodes. We will provide a snapshot of the design patterns and pattern languages that have been applied or that we plan to apply to a middleware framework we are developing for the Boeing NEST Open Experimental Platform (OEP) and their implications for NEST systems.

1830 - 2030 *Workshop Reception hosted by IONA Technologies*

THURSDAY, January 10, 2002

0830 - 1030 *Case Studies*

Chair: Jon Siegel, Object Management Group

Each of the Case Study presentations in this session has a little "something extra" in terms of technology. The first paper, Reliable UDP Transport for CORBA, tells how Lockheed-Martin developed a new transport system in order to provide the deterministic end-to-end behavior they needed for the Aegis Weapon System. The next paper, Model Driven Multi-Protocol Embedded Agents, describes a framework that manages the various aspects of a modern telecommunications architecture. A third paper, not yet listed, will show how OMG's new Model Driven Architecture (MDA) can be used to develop real-time systems.

Reliable UDP Transport for CORBA
Voula Fotopoulos and Catherine Heaberlin, Lockheed Martin - NE&SS-Surface Systems

The Aegis Weapon System (AWS) is transitioning to an open, standards-based, distributed, component processing architecture. This transition will provide the foundation for successful design and deployment of future combat system capabilities. Commercial middleware, such as RT CORBA, provides desirable distributed communication capabilities for large-scale real-time applications. However, the properties of TCP/IP in standard CORBA, such as deferred transmissions, delayed acknowledgments, and recovery mechanisms, can cause excessive overhead and latency that obstruct real-time predictability. This presentation discusses the efforts associated with the implementation of a reliable UDP/IP transport as an alternative, presents performance results, and shows that an ORB with a reliable UDP transport is suitable for supporting some real-time applications.

Model Driven Multi-protocol Embedded Agents
Mani Krishnan, Vertel Corporation

Carriers/Service providers are going through a major upgrade of their networks to offer enhanced services as Convergent Technologies are deployed. Carriers need to support network equipment from multiple vendors with multiple protocol interfaces. Equipment vendors have an increasing need to provide multiple interfaces to the embedded agent software, so that any carrier management system can access the network equipment. This paper will discuss the challenges of developing a model driven, multi-protocol, scalable, distributed, transaction-oriented engine for enabling various services for FCAPS (Fault, Configuration, Accounting, Performance, Security) management in the embedded platform.

Case Study: minimumCORBA ORBs in L4 Router
Jihoon Jeong, ROCOZEN Co., Ltd.

We have implemented minimumCORBA ORBs in an L4 (OSI layer 4) router to monitor and log machine status. OMG's minimumCORBA is light enough to replace SNMP protocol stack in the network equipment area. FLB-100 of FusionX is a popular router supporting either ADSL/CABLE network or IP sharing, which has been supplied over 2,000 game rooms. This router has only 16M RAM and no hard disk. ROCOZEN has ported miniSORBA to this router and provided NMS using SES 2001 (SORBA Enterprise Server 2001), java edition, verifying that minimumCORBA can replace existing SNMP protocol's role and provide more services than SNMP.

Model-Driven Architectures and Executable UML
Chris Raistrick, Kennedy Carter Limited

This presentation will illustrate, using examples drawn from a successful project, how the key principles of the Model Driven Architecture (MDA) and UML Action Semantics can be applied in the context of a system development. It will show how the UML Action Semantics standard supports construction of an executable model of the system, allowing the model to be tested at an early stage. This, in combination with the principles of the MDA allows us to decouple the model of required behaviour, expressed in the Platform Independent Model, from the automatically generated, optimised implementation.

1030 - 1045 Morning Refreshments

1045 - 1215 ***Transport Issues***

Chair: Shahzad Aslam-Mir, Vertel

The performance and behaviors of any ORB over transports other than TCP/IP vary enormously. Careful examination and characterization of these effects is therefore paramount if one is to apply such constructs in a way that is beneficial. Typically, the goals cited for switching to alternate transports have been to increase performance, or achieve greater determinism. However, the idea of using alternate transports brings with it several issues that need to be adequately understood and addressed. An alternate transport brings with it, not only unique benefits, but also a different set of constraints. Navigating this new, and possibly different, constraint set from that of TCP, in a manner that helps the ORB capitalize the transports benefits is something that requires some considerable study. This session presents papers that serve to highlight a small cross-section of such issues.

Mapping IEC 61850 to CORBA

Ricardo Sanz, Universidad Politécnica de Madrid

The IEC 61850 (Communication Networks and Systems in Substations) is an emergent standard from the IEC (International Electrotechnical Commission) that addresses interoperation issues in substation automation systems. The objective of the standard is to design a communication system that provides interoperability between the functions to be performed in a substation but residing in equipment (physical devices) from different suppliers, meeting the same functional and operational requirements. Systems built using this standard will be able to interoperate easily and even support dynamic configuration and function migration from embedded device to embedded device. In this paper we will describe the work done in the IST DOTS project developing a mapping between IEC 61850 and CORBA and implementing the 61850 models using a RT-CORBA infrastructure specifically tailored for industrial environments.

Managing Communications Bandwidth with a Real-time QoS ORB

Cathleen Hrustich, Objective Interface Systems

Middleware technologies have largely ignored the specialized parameters offered by communications quality of service (QoS) for controlling priority, bandwidth, jitter, and latency. Conventional Real-time ORB implementations do not allow access to, much less manipulation of, these QoS parameters. This presentation will identify a full CORBA real-time implementation providing full access and control of these QoS parameters, and the impact on the system when these parameters are controlled. Additional discussion will show how latency and jitter can be avoided by the application engineer, how replaceable transports can define new classes of QoS for application developers and how the selection of an efficient communications transport can have the greatest impact on QoS parameters.

Performance of ORBs on Switched Fabric Transports

Victor Giddings, Objective Interface Systems

Switched fabrics consist of parallel, high-bandwidth, and redundant data transfer paths between switching elements and host computers. Data is transferred between source and destination endpoints in dedicated paths through the switched fabric that offer low latency, high bandwidth, and low jitter. The performance characteristics of an ORB operating over a switched fabric transport are different from an ORB operating over Ethernet because of the increased number of processing elements involved. The host CPU, the switch elements, the fabric interconnect, and the host memory bandwidths all contribute to a more complex performance picture.

1215 - 1315 Lunch

1315 - 1445 ***Performance & Quality of Service***

Chair: Mark Gerhardt, TimeSys

Traditional timing performance requirements have been typically characterized by computational loading and its related completion deadlines. Recent characterizations of timeliness have produced additional qualitative and quantitative characterizations of architectures, components and behavior. Quality of Service deals with such concepts. Emerging work dealing with assessing componentry and frameworks by characterizing their behavior and Quality of Service will be presented in this session.

Benchmarking Real-Time and Embedded CORBA ORBs: Beyond the Obvious
Brad Balfour, Objective Interface Systems

When faced with the task of selecting an ORB for their embedded or real-time application, the first thing that most developers do is find out how fast the ORB is. Not that there's anything wrong with that… However, there is a lot more information that can, and should, be gathered. Much of it is unique to the analysis of embedded and real-time CORBA and won't be found by studying benchmarks of desktop and enterprise ORBs. Additional areas to be explored include: Measuring Multiple Characteristics of a Single ORB, Measuring the ORB vs. the Communication Layer, and Measuring the Effect of the ORB's Control of the Communication Layer.

Measuring Priority Inversions in Real-Time ORBs
Steve Grimaldi, Objective Interface Systems

There are a number of factors that affect the predictability and determinism of a real-time CORBA implementation. One factor is the presence of a priority inversion, and the impact it has on the overall system. A priority inversion occurs when the highest priority task is enabled, but not executed. The presence of an inversion may be caused by the ORB implementation, the operating system it is hosted on, the communication transports used, or the application logic. This presentation will provide an overview of priority inversions, how priority inversions can be avoided within an ORB, and how to accurately measure the effects of an inversion.

Using Quality Objects (QuO) Middlewarefor QoS Control of Video Streams
Craig Rodrigues, BBN Technologies

This talk will focus on a different middleware based approach for insuring end-to-end QoS, using Quality Objects (QuO) middleware developed at BBN Technologies. QuO provides the distributed application developer a rich toolkit for specifying QoS policies and requirements in the form of QoS contracts written in specialized Aspect Oriented Languages. QuO also provides the developer a set of System Condition Objects (Sysconds) which can provide information about system state to the QoS contract, and can interface with resource managers. QuO works in conjunction with middleware based on CORBA or Java RMI. The presentation will describe the work done to date by BBN, and provide some of our experimental results.

1445 - 1500 Afternoon Refreshments

1500 - 1700 ***Vendors' Roundtable***

Moderators: Richard Soley, Object Management Group
Jishnu Mukerji, Hewlett-Packard

This panel of vendors of embedded CORBA implementations will focus on the products and plans of the companies represented. Panelists will briefly introduce their products, product plans, standards conformance plans and suggestions for future standardization in this area. This will be followed by an open discussion with all workshop participants on the topic.

Panelists: Shahzad Aslam-Mir, Senior Architect , Vertel Corporation

Bill Beckwith, CEO, Objective Interface Systems

Ken Black, Founder and CEO, Highlander Engineering

Peter Kortmann, President, Tri-Pacific Software Inc.

Stephen J. Mellor, Senior Vice President, Project Technology, Inc

Others TBD

Program Committee

Chair: Richard Soley, Object Management Group

Members:

Dock Allen, MITRE
Shahzad Aslam-Mir, Vertel
David Barnett, Highlander Engineering
Bill Beckwith, Objective Interface Systems
Mark Gerhardt, TimeSys
Victor Giddings, Objective Interface Systems
Chris Gill, Washington University
Janice Gilman, Object Management Group
Michi Henning, IONA Technologies
E. Douglas Jensen, MITRE
Ji-hoon Jeong, ROCOZEN
Kane Kim, UC Irvine
Hermann Kopetz, Technische Universität Wien
Peter Kortmann, Tri-Pacific Software
Kevin Loughry, Object Management Group
David McKinnon, Washington State University
Jishnu Mukerji, Hewlett-Packard
Priya Narasimhan, Carnegie Mellon University
Doug Schmidt, UC Irvine
Bran Selic, Rational Software
Jon Siegel, Object Management Group
Rob Van Blommestein, Highlander Engineering
Andrew Watson, Object Management Group
Ben Watson, Tri-Pacific Software
Lothar Werzinger, KRONES