

“Not Your Father’s CORBA - An Architecture for Embedded and Real-Time Systems

The new CORBA/e is the high-performance architecture of choice for distributed real-time and embedded environments.

By Victor Giddings, Objective Interface Systems

Scientists at Lawrence Livermore National Laboratory’s National Ignition Facility (NIF) needed to solve one of the world’s most challenging distributed system control problems for a \$1.2 billion nuclear fusion facility. They required a high-performance systems architecture to enable maximum code reuse, accelerated performance and real-time application integration.

The NIF’s Integrated Computer Control System (ICCS) had to manage more than 40,000 control points to operate 192 of the world’s most powerful laser beamlines. These lasers deliver 2 MegaJoule pulses of optical energy onto a BB-sized deuterium fusion fuel capsule in a pulse 25 nanoseconds long, which results in an energy-producing inertial confinement fusion reaction similar to the ignition of a small sun. For this demanding engineering task, NIF depends on the Common Object Request Broker Architecture (CORBA).

Yes, that’s right. CORBA. This might surprise those who remember CORBA from its earlier incarnations in the enterprise software industry when it was typically characterized as “big, fat and slow.”

The development of real-time and embedded CORBA offers major improvements over earlier implementations of CORBA. Its performance meets or exceeds users’ most stringent requirements and allows programmers to develop source code that is reusable across multiple platforms, applications and projects—which maximizes an investment in CORBA across generations of applications.

Today CORBA has become the middleware of choice for the world’s most demanding applications and harshest environments in defense, aerospace, communications, banking/finance, industrial controls, robotics, medical equipment and multiple other industries.

What is CORBA/e?

For systems that need small memory footprint and deterministic execution, embedded developers can use the latest generation of CORBA: CORBA/e (CORBA for embedded). An Object Management Group (OMG) standard, CORBA/e provides an architecture for distributed processing that fits systems from the largest server farms to the smallest networked Digital Signal Processors (DSPs).

The OMG has merged the static aspects of industry standard CORBA with the essential Real-Time CORBA features into two new profiles grouped under the banner of CORBA/e. The CORBA/e Compact Profile fits easily on a typical 32-bit microprocessor, running a standard Real-Time Operating System (RTOS); these systems may run such applications as communications, signal or image processing with real-time dependability. The CORBA/e Micro Profile is even smaller and fits on the kind of low-powered microprocessor or high-end DSP found on mobile or handheld equipment.

Why CORBA/e?

Embedded system software development is an expensive and time-consuming task. But with a sound middleware architecture, this investment can pay dividends across many generations of technology. For developers of real-time and embedded systems, CORBA/e is ideally suited to the challenges of today's mission-critical environments.

In today's world, stand-alone systems are becoming a thing of the past. Embedded processor environments are networked and highly interconnected. Software must cope with communications and interoperability issues, while delivering the same reliability and performance as the isolated embedded systems of the past. Even systems that appear to be stand-alone need a communications infrastructure to merely report their status. CORBA/e provides easy access to a variety of sophisticated transports including Shared Memory, RapidIO and FireWire, as well as Ethernet.

Embedded software development is no longer confined to a specific processor model on a particular board—if it ever was. A developer may need to support many different processors at the same time, or migrate to new environments as the initially targeted compilers, operating systems and processors change. There is increased pressure to preserve investment through development of “reusable product lines.” Developers writing their own infrastructure are more susceptible to increased cost due to code changes accommodating inevitable environment changes. CORBA/e insulates embedded developers from the headaches of rewriting code with every processor and system change. In addition, embedded systems will increasingly use multicore processors, and developers will need to migrate their applications from single core processors to processors containing two or more cores. CORBA/e enables a seamless migration to multiple cores through the benefits of location transparency.

With this kind of increase in complexity, applications gain flexibility through refactoring. Developers can build a CORBA/e application as if it were a stand-alone application. They can then spread and re-spread the application logic across multiple resources in a way that doesn't create any extra work. Deferring deployment decisions enables optimized resource allocation and the flexibility to adjust to changing conditions.

In the embedded world, as we all know, systems and devices interact with the real world in real time (see sidebar “CORBA/e Real-Time Predictability”). Devices require interactions that are predictable in time as well as in function. CORBA/e provides distributed predictability by recognizing and propagating priority in its own processing

and across the system. The real world also has constraints on power, weight, size and speed. CORBA/e is specifically designed to support board-based and networked systems with the smallest footprint and the highest performance requirements. CORBA/e focuses on providing just the most useful features of previous versions of CORBA, paring away those that bloat footprint and processing requirements.

With the increased use of embedded devices in applications that are mission-critical and/or involve the possibility of injury or death, reliability must be built in. Real-time and embedded developers are rightly skeptical about adopting code they don't write themselves. Robust implementations of CORBA/e, field-tested in the harshest environments, have proven their reliability time and time again. Portability among implementations ensures the longevity of the solution.

And, of course, there are always economic pressures requiring greater productivity from systems development, so time-to-market is critical. By supplying a high-performance communications framework, CORBA/e enables greater productivity because it is no longer necessary for developers to write their own protocols. CORBA/e solves the tedious and time-consuming part of distributed applications by establishing a reliable, flexible architecture.

To the embedded system architect, CORBA/e levels the interoperability playing field, allowing board-based systems and micro systems to interoperate not only with each other, but also with existing systems using the CORBA standard. At the same time, it allows developers to use their existing CORBA skills to code at full speed from the start and allows projects to hire from a large pool of skilled CORBA developers.

CORBA/e systems are fully interoperable and support the OMG's mature interoperability standards: GIOP (General Inter-ORB Protocol) and IIOP (Internet Inter-ORB Protocol). In addition, some CORBA/e implementations also offer additional transports tuned to embedded targets and allow embedded developers to plug in their own custom transports. This flexibility allows embedded devices to interoperate with systems ranging from the largest server array installations to small chip-based sensors.

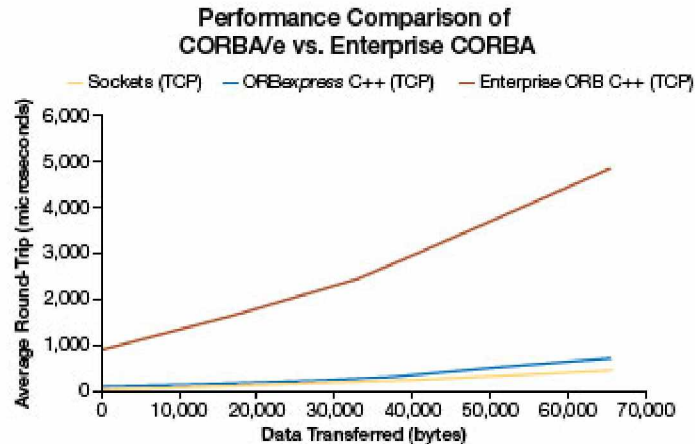


Figure 1 The minimal latency added by a robust CORBA/e implementation compared to a traditional enterprise ORB.

CORBA/e Profiles

The CORBA/e profiles—Compact and Micro—package the static features of CORBA middleware and real-time capabilities into a small footprint. Figure 2 shows the relationship between the CORBA/e profiles and standard CORBA. Designed by the most experienced providers of Distributed, Real-time and Embedded (DRE) software, both profiles fulfill the range of industry requirements.



Figure 2 The CORBA profiles for a continuum of size, interoperability and compatibility.

The CORBA/e Compact Profile merges the key features of standard CORBA 2.6 and Real-Time CORBA into a powerful yet compact middleware package that interoperates with other CORBA clients and servers of every scale, executes with the deterministic characteristics required of a true real-time platform, and leverages the knowledge and skills of software development teams through industry standard architecture. Shedding the dynamic aspects of CORBA and support for the CORBA Component Model (with their unpredictable response times and unlimited potential memory usage), CORBA/e Compact Profile retains interoperability by retaining the full IIOP protocol. It also preserves many server-side implementation options through the Portable Object Adapter

(POA) policies, and a rich type model that includes a lightweight version of “Valuetypes” and Type Any.

For even more compact systems, the CORBA/e Micro Profile shrinks the footprint even more, small enough to fit low-powered microprocessors or DSPs. This profile further trims features and constrains options that increase footprint, such as the Valuetype, the Type Any, most of the POA options, and all of the real-time functions except the real-time Mutex interface. In exchange for these limitations, the profile defines a CORBA executable that vendors have fit into only tens of kilobytes of memory—small enough to fit onto a high-end DSP or microprocessor on a handheld device. Even at this small size, the CORBA/e Micro Profile retains full IIOP interoperability (Figure 2).

CORBA/e middleware is available now. Based on vendor experience with military, industrial and consumer DRE applications, CORBA/e blends the standard and real-time specifications that the industry needs (see sidebar “Technical Features of CORBA/e”).

CORBA, as a standard, has continued to evolve over the last decade, and is adopted by many industries as the preferred middleware solution as evidenced by the large industry involvement within the OMG. Some of the most exciting standards development has come from the Real-Time Embedded and Specialized Systems (RTESS) Task Force in the form of CORBA enhancements designed for the real-time and embedded developer. A team of diverse companies, including representatives from telecommunications, aerospace and CORBA vendors, jointly authored the CORBA/e specification. Objective Interface was a key contributor in this initiative, taking a leading role in driving the specification to completion.

CORBA/e offers an architectural solution to keep up with ever-changing technology—in processors, models and particularly communications bus types. CORBA/e lets developers protect their investment in development work despite rapidly accelerating change.

Objective Interface Systems
Herndon, VA.
(703) 295-6500.
[www.ois.com].