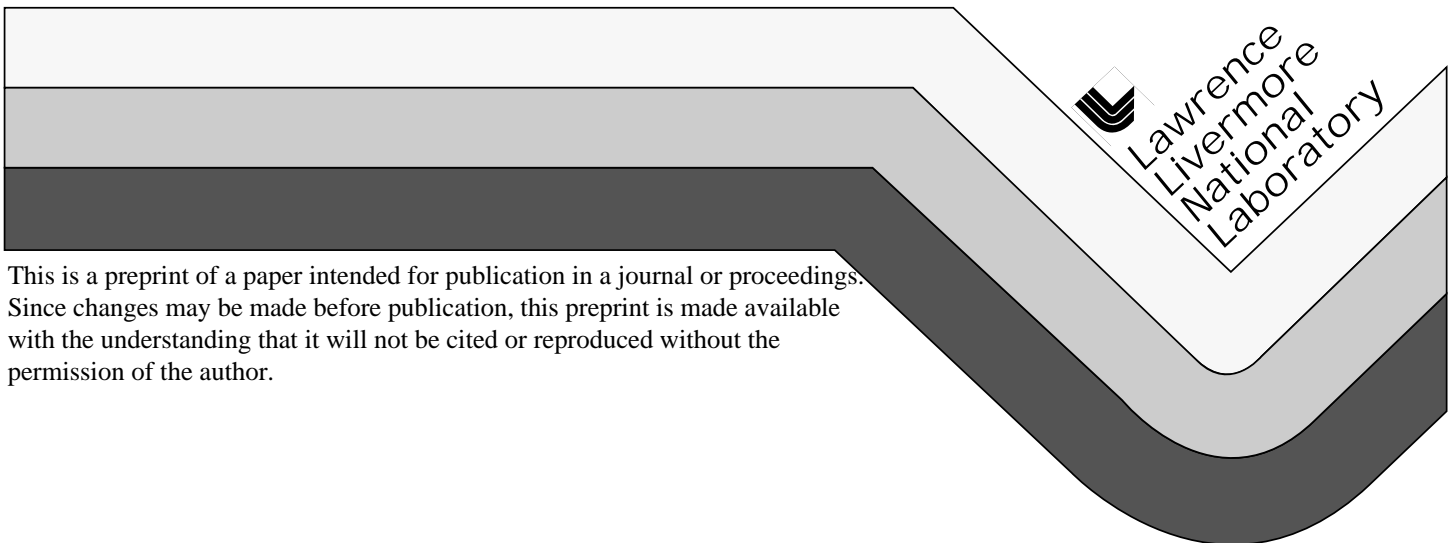


A Large Distributed Control System Using Ada in Fusion Research

John P. Woodruff

This paper was prepared for submittal to
Rational Software User Conference
Orlando, Florida
June 15, 1998

April 21, 1998



DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

A Large Distributed Control System Using Ada in Fusion Research

Abstract

Construction of the National Ignition Facility laser at Lawrence Livermore National Laboratory features a large distributed control system constructed using object-oriented software engineering techniques. Control of 60,000 devices is effected using a network of some 500 computers that run software written in Ada and communicating through CORBA. The project has completed its final design review; implementation of the first of five planned increments will be delivered at the end of fiscal year 1998. Preliminary measures of the distributed controls performance confirm the design decisions reported in this paper, and the measurement and supporting simulation of full system performance continue.

The National Ignition Facility Project

The \$1.2 billion National Ignition Facility (NIF) laser [1] is under construction at Lawrence Livermore National Laboratory in California. When completed in 2003, it will be housed in a building the size of a football stadium—704 feet long by 403 feet wide by 82 feet high. This laser is the latest in a series of experimental machines used to study inertial confinement fusion: nuclear fusion reactions produced in a plasma of deuterium and tritium that is compressed by a burst of laser energy [2] [3]. The NIF laser, which will be the world's largest high power laser, will deliver 1.8 MegaJoule pulses of optical energy onto a BB-sized fusion fuel capsule in a pulse 25 nanoseconds long. In an experiment, the target will be heated to more than 100,000,000 degrees Celsius and compressed to a density more than 20 times that of lead.

The scientific data that NIF produces in support of the inertial confinement fusion program will support three diverse objectives. As a key component of the US Department of Energy's Stockpile Stewardship and Management Program, the NIF will enable the US to maintain its nuclear device stockpile without resorting to underground testing [4]. It will also collect preliminary data about fusion as an environmentally attractive energy source [5]. The ability to re-create conditions existing inside the sun and stars will significantly impact the science of astrophysics and high energy density physics [6].

NIF's Integrated Computer Control System (ICCS) must integrate more than 60,000 control points to manage 192 laser beamlines. Each shot of the NIF, which generates about 400 Mb of data, will require aligning all components of the laser so that all 192 beams propagate down 600-foot paths through their amplifiers and into the target chamber within 50 microns of their assigned spot inside the centimeter-scale target — and so that all beams arrive at the center simultaneously within 30 picoseconds. The alignment process will involve approximately 9500 stepper motors, every one contributing to the position of the

beam. This alignment activity is carried out by an automated system that analyzes some 3000 distinct images, measures image offsets, and commands motion of the stepper motors; only when a control loop fails to stabilize is it necessary for an operator to intervene.

Control System Requirements

Facilities such as the NIF represent major capital investments that will be operated, maintained, and upgraded for decades. The facility is, in a sense, a factory in which physics experiments are manufactured. Accordingly the computer and control subsystems must be relatively easy to extend as innovative experiments are planned. To assure long service life, the system must permit periodic replacement with newer technology. The system is being built using Ada on a modern object-oriented software framework that will be extensible and maintainable throughout the facility's life cycle.

The NIF controls must be highly automated and robust, and will operate continuously around the clock with an allowed downtime of 7.5 days per year for unscheduled maintenance. A brief summary of requirements follows.

- The ICCS provides the necessary controls for a dozen engineers to carry out coordinated activities on a machine composed of about 60,000 distinct control points;
- The control system coordinates several experimental cycles concurrently to allow different sections of the laser to be used independently;
- Data are collected to support operational and maintenance planning;
- Selected images from a population of 600 cameras can be displayed to an operator at ten frames per second;
- Automatic alignment of the entire laser, controlling actuators in 4000 control loops driven by image matching control, takes place in one hour;
- The operator's broad view of the entire machine's status is updated with a latency less than ten seconds;
- The system architecture must be flexible enough to absorb significant changes in requirements late in project construction;
- The system must be delivered on time and on budget.

Layered Architecture

Several hundred computers running 8 supervisory and 14 kinds of front-end processor (FEP) applications are required to control and supervise system operation. The supervisory layer, which is hosted on UNIX workstations, provides centralized operator controls and status, data archiving, and integration services; FEP's implement direct control of devices.

The supervisory control layer is a software system estimated to comprise 400 KSLOC. This layer provides a human interface in the form of operator displays, data retrieval, and processing that is used to coordinate control functions across laser and target area equipment. The ICCS supervisor is partitioned into eight cohesive subsystems, each of which controls a primary NIF subsystem such as beam control or power conditioning. Several databases are incorporated to manage both experimental results as well as data used during operations and

maintenance. The supervisory software is responsible for duties ranging from configuration and control sequencing to data processing and archival.

FEP units are constructed from VME-bus or PCI-bus crates. FEP software provides the distributed services needed by the supervisory system to operate these control points. This software performs sequencing, data acquisition and reduction, and device control. The architecture includes a standard way for FEP units to be integrated into the supervisory system by providing a common distribution mechanism, coupled with patterns for device configuration, control, and status monitoring functions. Functions requiring real-time implementation are allocated to software within an FEP or embedded controller. These real-time functions are located at the edge of the architecture and therefore do not require communication over the local area network.

The control points are sensors and actuators attached to interface boards plugged into the FEP backplane. In many cases, control points are handled by intelligent components that incorporate embedded controllers operated by small fixed programs. This firmware that runs in the embedded controller does much of the low-level work that would otherwise be allocated to an FEP. Example components implemented in this manner are stepping motor controllers, photodiode detectors, and power supply controls. Laser diagnostic sensors attach to FEP units by utilizing low-cost field bus microcontrollers.

Software Architecture of the ICCS

The ICCS software architecture was created to address the general problem of providing distributed controls for medium to large-scale scientific facilities that do not depend on significant real-time capability within the supervisory software. This absence of hard deadlines in the upper levels of the NIF system is due to the fact that there is no on-going physical process taking place in the laser. Rather, the shot time-line occurs over several hours and can be suspended if necessary. Within the two seconds immediately surrounding a 25-nanosecond-long laser pulse, control is turned over to a hardware timing system. Therefore client-server tactics are appropriate for the supervisory level of the ICCS.

The principle activities of the control system are preparation for a shot and examination of data from previous shots. The supervisory system provides access to the laser's subsystems for up to fourteen operators and for the experimenters. These services include interactive manipulation of numerous control points (usually coordinated into subsystems) and examination of data. Stored history of operational parameters and performance, as well as signals and images generated by NIF sensors are presented to operators.

Risks to the control system infrastructure are mitigated in the ICCS design by incorporating modularity, segmentation, and open-systems standards so that components and subsystems can be replaced at designated interface points if necessary. This strategy is being realized by building object-oriented software in the Ada language. And to address the significant cost, reliability and maintenance issues of a control system of this magnitude, a model-based, object-oriented approach to the design and development of the ICCS software was

chosen. The software engineering process uses models expressed in the Unified Modeling Language (UML) notation [7] [8].

Interoperability among computers and operating systems is addressed by leveraging the international standard Common Object Request Broker Architecture (CORBA) [9].

Some real-time control is inevitably necessary. Control loops requiring deterministic response are implemented at the periphery of the integrated architecture in front-end computer equipment. Programmable logic controllers in the industrial control subsystem handle safety interlocks and utility controls for thermal, gas, and vacuum. Diagnostic instrumentation and controllers that are activated within two seconds of the laser shot are triggered by an integrated timing system. In any event, no hard real-time control needs to be implemented over the computer system network.

Software Frameworks

The control system architecture is managed through a modern object-oriented software framework that is used to construct all applications and that will be extensible and maintainable throughout the project life cycle. This framework is reusable across multiple applications (and indeed other projects), and offers interoperability among computers and operating systems by leveraging CORBA.

The ICCS supervisory software framework delivers prebuilt components that are extended to accommodate specific additional requirements in the construction of the application software. The framework promotes code reuse by providing a standard model and interconnecting backplane that is shared from one application to the next.

The abstract components that are being constructed for NIF include

- Configuration — defines the naming convention used for the control points, forms an organization of the static data that define the hardware control points accessible to the ICCS, and is responsible for initializing the FEP's during startup.
- System Manager — provides services for the integrated management of the ICCS network of hundreds of computers, ensures that necessary processes and computers are operating, and provides failover services for processes that terminate during operation.
- Generic Front End Processor — adds unique classes for supporting device and controller interfacing.
- Status Monitor — provides generalized services for broad-view operator display of device status information using the push model of status change notification.
- Sequence Control Language — creates custom scripting languages for the NIF operation that are created and maintained by operations staff.
- Message Log — provides event notification and archiving services to all subsystems within the ICCS.
- Graphical User Interface — enables all human interaction with the ICCS, via GUI's displayed upon control room consoles or on terminals distributed throughout the facility.

- Reservations — manages access to devices by giving one client exclusive rights to control or otherwise alter the device.
- Machine History Archive — collects data about the performance and operation of the NIF, originating within the ICCS, for analysis of the NIF operation in order to improve efficiency and reliability.
- Shot Data Archive — collects the 400-Mb data generated from the diagnostics, makes it immediately available for “quick look” analysis, and delivers it to an archive.

Preliminary implementations of six of these framework components have confirmed the architecture and the use of Ada in its implementation. Proofs of principle that demonstrate how frameworks interact are described further on in the section “Measurements that Confirm Feasibility.”

Incremental Development Strategy

ICCS software development is managed under a software quality assurance plan that covers the entire life cycle of the software design, production, and maintenance. Supporting that plan are software engineering tools that have been proven capable of executing large object-oriented projects. The CASE tools selected include the Rational Rose modeling tool, the Rational Apex Ada compiler and configuration management. Objective Interface System’s ORBexpress distribution middleware and the Oracle 8 database management system both contribute to the planned system, as does a yet-to-be-finalized X-Windows based graphic interface construction tool.

Central to the development are documentation standards for software requirement specifications (SRS) and software design descriptions (SDD). These documents are essential to the long-term maintainability of the software in view of periodic software upgrades and staffing turnover expected during the 30-year life of the NIF.

Development is guided by an iterative approach to software construction. This technique is believed to be effective for projects whose requirements are not fully known until late in the project development. At the earliest stages of the project, decisions were made based on exploratory programming exercises. The team examined the risks posed by such untested techniques as model-driven object-oriented development, Ada-95, and CORBA communication by designing and testing very small subsets of the system.

Five iterations of product construction and testing are planned subsequent to project Final Design Review (FDR) and before deployment of the ICCS software. Each new release will follow a plan aimed at addressing the greatest risks to the architecture while increasing functionality.

The first of these iterations will show the application of already-prototyped frameworks to selected control system functions. This first production prototype implements so-called “vertical slices” of system behavior including both supervisors and FEP’s. Each subsystem will endeavor to demonstrate just one of the several planned shared frameworks: the pulse power conditioning

application will exhibit the shot count-down operation, automatic alignment will demonstrate the customizability provided by sequence scripts, and so on.

The second and third production prototypes will increase the degree of integration in the ICCS software, and will provide services to selected laboratory-scale tests using instances of the Generic FEP. In the middle of year 2000 a fully functioning control system will be deployed into the first phase of NIF: eight of the 192 laser beams will be installed and tested under ICCS control during that year.

Measurements that Confirm Feasibility

Two cycles of exploratory programming were carried out before the project FDR. At the completion of these preliminary iterations, approximately 5% of the estimated lines of code had been prototyped and two substantial development risks had been addressed. Object oriented techniques had been used to construct representative parts of ICCS in Ada, and these parts, as well as numerous standalone tests, had confirmed the ability of CORBA to support ICCS performance requirements.

The architecture was confirmed by the prototype. Several distinct kinds of devices were defined and elaborated by the configuration framework, including multi-axis gimbals implemented using stepper motors, photodiodes for laser energy measurement, and precision timing equipment. These device tests relied on data transmitted from an Oracle database into remote FEP's that were instantiated from the generic framework. The tests also demonstrated distributed control and monitoring provided by the architecture. Software infrastructures for system start-up and process management, for audit trail generation, and for database query were examined. NIF's automatic alignment design, which uses image analysis and closed-loop control to align optics without human intervention, was confirmed in an optical testbed.

Timing tests and load tests running on workstations like those specified for NIF show that CORBA can sustain a throughput of 800 control messages per second. A separate test confirms that digital images can share the network with the control system and that these images display on the console at a rate exceeding the required ten frames per second.

Continuing testing will continue through 1998, including the delivery of a production prototype to an independent testing organization for preliminary reliability estimation. Nearly all control functions will be deployed to control the first "bundle" of eight beams in early 2000, and experiments on that laser will be carried out late that year. Project completion will occur in 2003.

References

[1] J. A. Paisner and J. R. Murray "The National Ignition Facility for Inertial Confinement Fusion," 17th IEEE/NPSS Symposium on Fusion Engineering, San Diego CA, October 6-10, 1997.

- [2] E. Michael Campbell, Neil C. Holmes, Steve B. Libby, Bruce A. Remington, and Edward Teller "The Evolution of High-Energy-Density Physics: from Nuclear Testing to the Superlasers" *Laser and Particle Beams* 1997, vol. 15, no. 4, pp 607-626.
- [3] "Veil of Secrecy is Lifted From Parts of Livermore's Laser Fusion Program," *Physics Today*, September 1994, p 17.
- [4] Ted Perry and Bruce Remington "Nova Laser Experiments and Stockpile Stewardship" *Science and Technology Review*, September 1997
<http://www.llnl.gov/str/Remington.html>
- [5] William J. Hogan, Roger O. Bangerter, and Charles P. Verdon "The Fire Next Time" *The Sciences*, Vol. 36 No. 5, September/October 1996, p 20.
- [6] Keya Davidson "From Swords to Supernovae," *Sky and Telescope*, November 1997, p 36.
- [7] "UML 1.1 Specification" <http://www.rational.com/uml/>
- [8] James Rumbaugh, Ivar Jacobson, and Grady Booch "Unified Modeling Language Reference Manual" Addison Wesley, expected 1998.
- [9] Object Management Group "The Common Object Request Broker: Architecture and Specification" John Wiley and Sons 1995.

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551

