Available online at www.sciencedirect.com
ScienceDirect
Fusion Engineering and Design 83 (2008) 535–538
Fusion Engineering and Design
www.elsevier.com/locate/fusengdes

# CORBA-based distributed software framework for the NIF integrated computer control system☆

E.A. Stout*, R.W. Carey, C.M. Estes, J.M. Fisher, L.J. Lagin, D.G. Mathisen, C.A. Reynolds, R.J. Sanchez

*Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, USA*

**Abstract**

The National Ignition Facility (NIF), currently under construction at the Lawrence Livermore National Laboratory, is a stadium-sized facility containing a 192-beam, 1.8 MJ, 500 TW, ultra-violet laser system together with a 10-meter diameter target chamber with room for nearly 100 experimental diagnostics. The NIF is operated by the Integrated Computer Control System (ICCS) which is a scalable, framework-based control system distributed over 800 computers throughout the NIF. The framework provides templates and services at multiple levels of abstraction for the construction of software applications that communicate via CORBA (Common Object Request Broker Architecture). Object-oriented software design patterns are implemented as templates and extended by application software. Developers extend the framework base classes to model the numerous physical control points and implement specializations of common application behaviors. An estimated 140,000 software objects, each individually addressable through CORBA, will be active at full scale. Many of these objects have persistent configuration information stored in a database. The configuration data is used to initialize the objects at system start-up. Centralized server programs that implement events, alerts, reservations, data archival, name service, data access, and process management provide common system wide services. At the highest level, a model-driven, distributed shot automation system provides a flexible and scalable framework for automatic sequencing of workflow for control and monitoring of NIF shots. The shot model, in conjunction with data defining the parameters and goals of an experiment, describes the steps to be performed by each subsystem in order to prepare for and fire a NIF shot. Status and usage of this distributed framework are described.


*Keywords:* National Ignition Facility; NIF; CORBA; Integrated Computer Control System; Eric stout; Lawrence Livermore National Laboratory; Object-oriented software

## 1. Introduction

The Integrated Computer Control System (ICCS) is a distributed, layered, object-oriented control system that employs a framework of reusable software to build uniform programs to satisfy numerous functional requirements for NIF (The National Ignition Facility) [1]. ICCS applications are developed using Ada95 or Java, CORBA (Common Object Request Broker Architecture), and object-oriented programming. Ada is used to implement most of the control system semantics. Java is used for the production of graphical user interfaces as well as some of the central services; some new controls applications are also being written in Java. CORBA provides location- and language-transparent distributed communication utilizing TCP/IP transport.

NIF is comprised of 192 laser beams, which are divided into 24 essentially identical 8-beam "bundles." The control system computers and processes are likewise largely partitioned by bundle, with the exception of central services and applications controlling front-end, target area, and industrial controls systems that are not replicated. This partitioning is a key element in assuring the scalability of the control system.

The top layer of the ICCS bundle-based architecture is the shot automation system. Shot Director software manages the progress of a NIF experiment through a sequence of states. In each state, Collaboration Supervisors, one for each bundle and one for shared, non-bundle resources, manage the workflow of a sequence of steps defined in a shot model. These steps, in turn, are distributed to Subsystem Shot Supervisors for execution [2].

☆ This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

* Corresponding author. Tel.: +1 925 423 8863; fax: +1 925 422 1930.
*E-mail address:* stout6@llnl.gov (E.A. Stout).

0920-3796/$ – see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.fusengdes.2007.12.025

Operational experience and analysis indicates that ICCS will scale successfully to full NIF operations. A full-power 6-bundle shot was performed in late 2006, and preparations are underway for the first low-energy 12-bundle test.

## 2. Bundle-based architecture

The NIF physical organization lends itself to a distributed, component-based communication architecture. Control system processes and computers are organized by bundle to achieve better parallelism, performance, and reduce the impact of localized failures. This is referred to as "bundle based partitioning" and has no impact on framework or application layer software due to the location independent features of the CORBA-based inter-process communication architecture. This ability to independently organize the hardware architecture without impact to the software is one of the significant benefits of CORBA and the design of the ICCS software framework.

ICCS employs a layered, client-server computer architecture to control the NIF. Each NIF bundle is supported by ~25 front-end processors (FEPs) connected to various laser diagnostic and control components. Additional FEPs control and monitor common components that are not bundle-based, such as target area systems. Each bundle has a dedicated Unix server on which the supervisory and shot control applications for that bundle are run. Additional servers support central service applications and non-bundle based supervisors. The control room contains Windows consoles dedicated to functional subsystems; GUIs run on these consoles and connect to the supervisory and FEP layers as needed. At full scale, the ICCS will be distributed among approximately 800 computers running 1500 processes containing 140 thousand individually addressable CORBA objects.

Following strategies of object-oriented software development, similar software components are defined as classes, and these classes are instantiated for each occurrence of a NIF component in each laser beam. Control components consist of various actuators, sensors, and instruments used to operate and diagnose each NIF laser beam and its interaction with a target. NIF physical control components, as well as the supervisory objects that aggregate the status and control of those physical components, are represented by named CORBA software objects in the ICCS. The framework and application software combined have resulted in a design consisting of approximately 340 CORBA interface classes. The ICCS framework allows client software to obtain a CORBA reference to an object from its name, and CORBA transparently connects that reference to the software component object located in the processor that is connected to the physical hardware. The client need not have explicit knowledge of the process or processor on which the object is running. This gives computer/network location transparency to the ICCS application software.

There are a number of challenges to making such a distributed system robust and resistant to failure. A high number of distributed interfaces and objects performing concurrent, interdependent activities leads to non-deterministic messaging behavior and the potential for race conditions, distributed deadlock, or lost connections as a result of restarting a process. ICCS employs a variety of mechanisms under the broad heading of "connection management" to mitigate these failure modes, including interface decoupling, object reconnection, subscription management, process state heartbeats, status health heartbeats, and timed invocation. This framework approach has enabled successful detection, notification, and recovery from communication failures, providing common benefits to all ICCS application software.

## 3. Shot automation system

Fundamentally, this large, distributed system exists for the purpose of enabling frequent, reliable NIF shots in support of experimental goals. The stated requirement for ICCS is to fire and diagnose each laser shot within 4–8 h.

In a typical shot sequence, the laser is first setup to meet the experimental requirements. Automatic alignment software analyzes beam images and moves mirrors to align each beam accurately along the beam path and ultimately onto the target, and diagnostics are setup to obtain data from the shot. Then multiple low-energy "rod" shots, for which the power amplifiers are not fired, are taken, and settings adjusted based on the results, until the pulse shape and pre-amplified energy of every beam on the shot meet the experimental requirements. Finally, a high-energy "system" shot, in which the power conditioning system charges the main capacitor banks and fires the flash lamps that amplify the beams, is taken. Each shot is preceded by a 4-min countdown during which an automated sequence of final actions and verifications is performed, and followed by data acquisition, analysis, and archival. Three layers of shot automation software orchestrate this intricate sequence: a Shot Director, Collaboration Supervisors, and Subsystem Shot Supervisors.

Upon selection of a predefined experiment by the Lead Operator, the Shot Director initiates calculations to determine operational settings and participation status for all of the NIF's components. It then manages a high-level state machine that shepherds the Collaboration Supervisors participating in the experiment together through each phase of the shot sequence. Once all Collaboration Supervisors have completed their activities for a state, the Shot Director moves them to the next state, either automatically or at the request of the Lead Operator, depending on the state. Additionally, in the countdown and post-countdown phases, the Shot Director publishes clock ticks and manages any holds issued when problems occur. Two seconds before the shot (T-2), the Shot Director hands off control to the Integrated Timing System, which issues precise triggers to fire the laser so that all beams arrive at the target simultaneously.

There is one Collaboration Supervisor per bundle, plus two for non-bundle based systems. The Collaboration Supervisor is a workflow engine, coordinating all of the activities performed by the Subsystem Shot Supervisors in its area. When the Shot Director initiates a new shot state, the Collaboration Supervisor queries the database for the workflow model for that state. This model defines the high-level functions to be performed by each Subsystem Shot Supervisor, as well as their order and interde-