Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304
Phone +1.415.855.7400
Fax +1.415.856.6153
Web http://www.echelon.com/lns

ECHELON

The LONWORKS® Network Services (LNS) Architecture Strategic Overview

Client-Server Control

Jeff Lund

Rev 1.0

Document No. 39305

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA, 94304, USA

Introduction

The LONWORKS® Network Services (LNS) architecture provides the foundation for interoperable LONWORKS network tools. It brings the power of client-server architecture and component-based software design into control networks — enabling a new generation of tools that can work together to install, maintain, monitor, and control LONWORKS networks. And, it offers the fastest way to bring control on-line with all your other information systems.

This paper provides a high-level overview of the features and benefits of the LNS architecture and the opportunities that it enables for end-users, system-integrators, and OEMs . For a technical overview of the components that make up the LNS architecture see *The LONWORKS Network Services (LNS) Architecture Technical Overview*, Echelon document number 39310 and the LNS home page at `http://www.lonworks.echelon.com/lns`.

The LNS Architecture — The Benefits to End-Users, Integrators, and OEMs

The Power of Client-Server Architecture

The benefits of client-server architectures have made them the standard for data networks. Almost all corporate data networks employ a client-server architecture. The Internet — one of the hottest emerging markets — is probably the worlds largest client-server system. LNS brings the benefits of client-server architecture to control systems. It provides the control networking industry's first multi-client, multi-server network operating system, overcoming many significant limitations faced by competing bus systems.

LNS allows multiple system integrators, managers, and maintenance personnel to simultaneously access network and application management services and data from any number of client tools. What does this mean for you?

- *Reduced commissioning time and cost*. Older control networks restrict you to a single tool for system installation and maintenance, making installation a serial process. With LNS, multiple installers can work on the same system, at the same time, without conflicts. Each tool is a remote client to the network server, allowing multiple installers to work in parallel without database synchronization problems.

  And, since these client tools do not need to contain a network database[1], they can be anything from laptop PCs to inexpensive hand-held tools with low cost microcontroller hosts and simple LCD displays. In fact, each client tool can, if desired, have a different user interface, optimized to the particular network subsystem being installed (e.g., material handling, access control, gas analysis, HVAC) or to the skill set of the user. By building application intelligence into network tools, all or part of the commissioning process can be automated, further reducing commissioning time and training cost.

- *Simplified system integration.* LNS applications have the option to provide their own set of application services, properties, and events and to access those provided by others. LNS handles the directory management and routing transparently to pass requests from clients to application servers. This allows systems to communicate with one another at a system-to-system level as well as device-to-device.

  For OEMs, this offers another way for them to add value to their system by embedding application intelligence into their tools. For integrators, system-to-system communication reduces the need to understand the implementation details within a given system. And, for end-users it results in disparate systems that work together, leading to more efficient operation, higher yield, and greater comfort.

[1] Client tools only need to contain a *Network Services Interface* (NSI), a low-cost hardware device that provides transparent access to the services and data provided by LNS network and application servers.

- *Greater access to data*. Whether its greater access to process data to increase yield and quality or better access to operating data to lower cost, better access to data increases efficiency and reduces cost.

  But data is often trapped in different devices, systems, and subsystems. Information *creation* is insufficient to give competitive organizations a strategic advantage—they need *access* to the data. LNS provides this access; it gives users the ability to deploy any number of MMI, SCADA, and data logging stations. Because of its client-server architecture, there are no databases to copy or redundant updates to make; users no longer worry about tools losing synchronization with the network's configuration. LNS tracks the needs of each tool and automatically informs them of configuration changes.

- *Increased system up-time*. When it comes to system maintenance, a single tool is never in the right place. Its always in the supervisor's office or on the other side of the plant nowhere near the fault. With LNS, repair technicians can plug tools into any point in the network and access all network services and data. Since multiple tools can interoperate on the same network, multiple technicians can be diagnosing problems and making repairs simultaneously with no need to coordinate their actions or to even be aware of one another. By building application intelligence into their tools, OEMs can further reduce system downtime by automating fault detection, isolation, reporting, and repair.

- *A path for growth*. Because of its client-server design, LNS allows smooth system growth. Through modular hardware and software components, users may expand their control systems by simply adding additional clients or servers — at any time. To the end-user, the change is transparent except for the increase in capacity and functionality.

The Power of Component-based Software

Object-based software design is quickly becoming the standard for software development. With the LONWORKS Component Architecture (LCA) extensions, LNS provides a standard ActiveX™ (formerly called OLE) interface to LONWORKS networks. What does this mean for you?

- *Faster software development*. LCA defines a standard object layer and set of interfaces. This allows developers and systems integrators to quickly assemble PC-based applications such as MMIs, SCADA applications, and configuration tools using off-the-shelf software components from multiple vendors. These components may run on the same host PC or interoperate over the LONWORKS network. LCA also lets developers take objects created for one application and reuse them in another.

- *Better products*. LCA gives developers the ability to focus only on those pieces of the application where they add the most value and to assemble the rest from best-of-breed components from other vendors.

- *Lower maintenance cost*. Because all implementation detail is hidden from other objects (other software modules), it is easy to modify an object's internal details without affecting any other objects in the system. As a result, object-based systems are much more flexible and easier to maintain than their procedural-based counterparts. In addition, because objects can only be accessed through well-defined interfaces, called *methods*, they are very hard to misuse. This results in fewer errors.

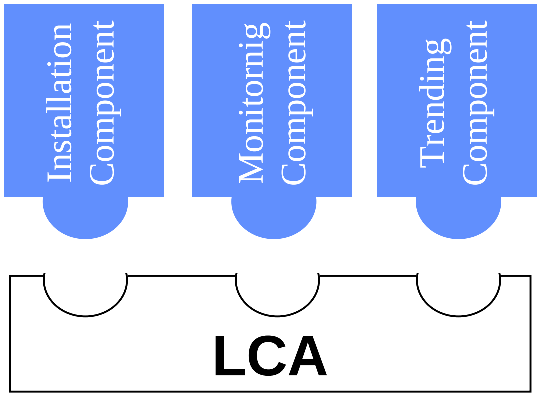

Figure 2. The LonWorks Component Architecture provides a standard software backplane into which users can plug intelligent components to assemble network tools

Technology Overview

LNS provide a compact, object-oriented programming model that reduces development time, host code space, and host processing requirements. LNS represents the network as a hierarchy of objects that provide a set of services, contain a number of properties, and that report changes with events. To simplify development wherever possible, LNS automates common system tasks. For example, it automatically discovers the presence of new unconfigured nodes on the network — without a tool's host having to do anything. LNS manages the network, freeing developers to focus on their application.

The Key Components — Network Services Servers and Interfaces

The LNS architecture consists of two major components, the *Network Services Server* (NSS) and the *Network Services Interface* (NSI).

The NSS hosts and processes network services. It maintains the network database. It enables and coordinates multiple points of access to its services and data. It also maintains a directory off all network and application service providers and event generators.

Clients "speak" to servers via a hardware component called the Network Services Interface (NSI). NSIs provide the physical connection to the network and the messaging connection to the NSS. They automatically consult the NSS as needed to determine which server provides a given service and route the request transparently.

Open To Any Host

The LNS architecture is host-independent, supporting clients on any platform, including embedded microcontrollers, Windows PCs, and UNIX workstations. LNS servers are based on Windows 95, Windows NT, or the Neuron Chip. LNS applications interact with the LNS infrastructure through the LNS host API[2]. The same host API is used by all clients; it does not matter what the host processor or operating system is. The API also presents a common interface for local and remote access to LNS services. To client applications there is no difference, all service request and result routing is handled transparently by the LNS infrastructure.

[2] The LNS Host API is a thin layer of code provided as portable ANSI C source in LNS Developer's Kit for Microcontrollers and the LNS FASTART Package. A 32-bit Windows implementation is provided in the LNS Developer's Kit for Windows and the LNS FASTART Package.

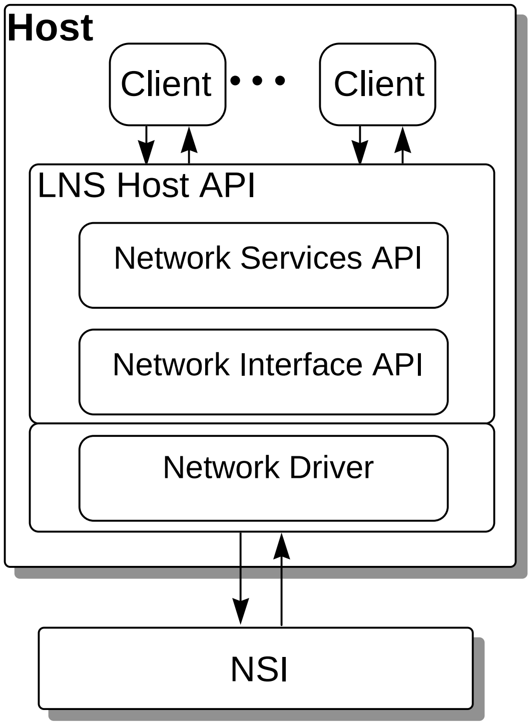

Figure 5. *Structure of the LNS Host API*

Leveraging Windows 95 / NT

On Windows NT and 95 platforms, LNS leverages Microsoft's OLE and ActiveX technologies — the Windows standards for component-based software. For these hosts, LNS defines the LONWORKS Component Architecture (LCA)[3]. LCA represents LNS objects as Microsoft ActiveX (OLE) objects, decreasing development effort by allowing you to leverage all support for ActiveX controls built into Windows development tools.

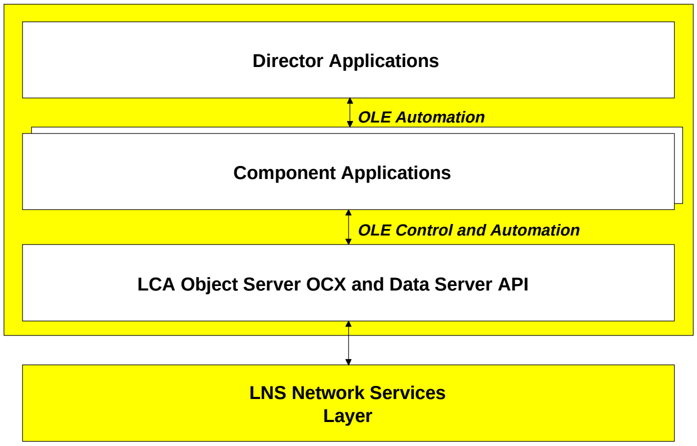
Director Applications
OLE Automation
Component Applications
OLE Control and Automation
LCA Object Server OCX and Data Server API
LNS Network Services Layer

Figure 2. The LCA Framework

By representing LNS objects as OLE objects, the LCA Object Server OCX greatly simplifies application development. OLE controls are high performance, 32-bit, language-independent programmable objects that can be used with a variety of development tools, including rapid application development (RAD) tools such as Visual Basic,

[3] The LCA Object Server OCX and LCA Data Server API are provided in the LNS Developer's Kit for Windows and the LNS FASTART Package.

as well as full featured programming environments such as Visual C++. These tools make application development with OLE controls simple because the OLE controls are displayed as icons in a toolbox that are simply dragged onto a form. OLE containers can therefore be implemented without knowing the underlying OLE and COM mechanisms.

The object server is a network tool kernel that enables sharing of information and objects between multiple tools and components. The kernel also provides an extensible host database that contains host-specific data not managed by the NSS as well as application-specific data.

To simplify development of Windows-based MMI and SCADA application, LCA provides a high-performance engine for system-level monitoring and control. This engine converts raw network data into formatted text strings or binary data which can be directly displayed and, optionally, filters redundant updates so that only data changes are reported.

LNS Products

LNS Developer's Kits

LNS developer's kits provide a comprehensive set of tools for rapid development of system-level tools for the host processor of your choice.

LNS FASTART Package	Complete set of tools, technical support, and hands-on training for developing LNS applications on any host — includes the tools from both the LNS Developer's Kit for Microcontrollers and LNS Developers for Windows
LNS Developer's Kit for Windows	Complete set of software components and technical support required to build LNS applications on Windows NT and Windows 95 hosts
LNS Developer's Kit for Microcontrollers	The hardware and software components and technical support required to build LNS applications based on any microcontroller, microprocessor, or computer host
LCA Field Compiler API	Windows libraries for creating field programming and debugging tools for application devices
NSS-10 OEM License	License to the NSS-10 hardware design and binary image to enable construction of custom form-factor NSS-10 modules

Embedded NSS Options and NSI Options

LNS OEM components provide hardware interfaces for building stand-alone embedded server applications as well as multi-client applications based on PC and embedded hosts.

NSS-10 Module	Single, local-only, host LNS server in a single in-line module that supports up to 62 nodes
PCNSS Interface Card	Half size ISA compatible PC board with an embedded NSS-10 server module
NSI-10 Module	Client interface in a single in-line module
PCNSI Interface Card	Half size ISA compatible PC board with an embedded NSI-10 client module

PCC-10 Network Interface	Type II PC, formerly PCMCIA, card NSI interface with integral free topology transceiver

For More Information

For more information on the LNS architecture, see the LNS home page at http://www.lonworks.echelon.com.

In addition, the following papers are available describing various aspects of the architecture. These papers are all available on the LNS home page.

- *The LonWorks Network Services (LNS) Architecture Technical* Overview, document 39310.
- *Developing Interoperable Applications and Systems with LNS*, document 39311.
- *Developing a Network Tool for Windows Using LNS*, document 39312.
- *Developing Device Type Specific Software for Windows with LNS*, document 39313.
- *Developing a Hand-Held Network Tool with LNS*, document 39314.
- *Managing Small Systems with LNS*, document 39315.