Controlling Audio Systems With ActiveX Controls Over CobraNet And Other Ethernet-Based Networks

Stephen R. Macatee and Devin Cook

Rane Corporation
Mukilteo, WA 98275-5098 USA

Networked audio systems are the present and future of the professional audio industry. Use of Microsoft ActiveX controls to control multiple CobraNet devices from one or more PC locations over an Ethernet network are discussed. Due to the pro audio industry's understandably conservative transition into such a future, a means to also incorporate current, familiar and trusted equipment into networked systems is a worthy endeavor. This is necessary until networked equipment is more prevalent, understood and trusted. A means to control such legacy equipment and to incorporate tools such as relays, switches and indicators into networks is also demonstrated using ActiveX over standard CobraNet or any Ethernet network. The biggest advantage to the ActiveX approach is the ability to control many manufacturers' devices from a single user interface while needing no cooperation between manufacturers.

I. INTRODUCTION

Control of audio systems and transport of audio over computer networks are newsworthy topics in the professional audio industry. Several schemes are available today. Using ActiveX is not new to the audio industry, but controlling CobraNet devices with ActiveX is. The secondary problem of incorporating real world relays, switches, indicators and existing non-networked, serially-controlled audio and interface products needs addressing to transition into our future. Also, the ability to link on-screen controls to such relays, switches, indicators and equipment must be supported.

To implement such control over Ethernet networks, Microsoft ActiveX controls have been employed as a control implementation. A brief primer on ActiveX as well as ActiveX advantages and disadvantages are discussed. With years of dedicated work behind the SC-10 and current AES-24 committees and the old topic of true cross-manufacturer interoperability still not generating the critical mass it once was, a view toward these topics is also revisited. This paper attempts to show that the nirvana of true manufacturer interoperability – regardless of protocol – is achievable over networks without the need for manufacturers to agree – except on ActiveX.

II. BACKGROUND

Much has been written on the varying needs, implementations, benefits and pitfalls of system control. The Audio Engineering Society (AES) has held conferences and paper sessions on computer-controlled audio systems.[1] With the seemingly constant drop in price and availability of computers, it is easy to anticipate further and even ubiquitous incorporation of computers as the control platform of choice for professional audio systems. Thousands of systems already incorporate personal computers, Macintosh and other computers.

The natural evolution of audio and control over standard Ethernet computer networks already taking place furthers the need to enable disparate computer-controllable devices to share both a common user interface and the transport mechanism over Ethernet. Peak Audio's CobraNet technology offers one such solution. Others to watch for increased applicability in the pro audio world include the currently consumer and professional audio-deployed IEEE 1394 (FireWire), and the recently released and demonstrated USB 2.0. ATM and other technologies from the computer industry will continue to infiltrate the conservative pro audio world.

The complexity of audio systems continues to grow, driven by users' requirements and expectations and the constant march of technology. The need for untrained users to easily control such complex systems is perhaps the hardest thing to provide when equipment from multiple manufacturers is incorporated into a single system.

Control interfaces began with device front panel controls and evolved through the following: wired and wireless remotes, MIDI, RS-232, RS-422, RS-485, DMX512, etc. The list continues through Ethernet and many other proprietary and non-proprietary schemes. Yet, with all these control and several transport schemes to choose from, it is still difficult to control disparate products from a common user interface and have the results work cohesively.

The dream of a common hardware and software approach permitting cross-manufacturer interoperability has lost a lot of steam since the Lone Wolf days of the pro audio industry. Much is learned from examining where we've been; where we're going; how we got here; why we were going there in the first place and what mistakes were made along the way.

The Lone Wolf dream of linking all manufacturers' products together with a common user interface and transport mechanism created great excitement and anticipation. The idea was simple, but the implementation proved too difficult. The Lone Wolf dream failed due to sale of the technology before it worked well enough for the scaleable needs of the industry, as well as the sudden licensing of too many manufacturers before the resources to support them were in place. Since then, the AES SC-10 and subsequent AES24 committees have continued the dream by working toward an agreement among audio manufacturers on a common communications protocol. Their work continues but is more likely to take a different approach. One thing is certain, Lone Wolf and SC-10 succeeded by providing the audio industry with a vision and the facility for cooperation toward that vision. It is now taking the technology of the computer industry to get us there though.

The need for a common communications and transport protocol applicable across disparate components from many manufacturers was achieved by the computer industry many years ago. It is often discussed that many standards come about through the necessity of one or more entities in an industry having the initiative to solve the problem at hand. Then others recognize its significance and jump on the bandwagon thus creating the momentum for a future standard, and/or competitive alternate solutions are developed. This is perhaps where the current work of AES24 lies – taking working examples and developing them as standards. In this case, as is happening in many industries, adopting techniques and technology from the long-standing and significantly larger computer industry with its enticing economies of scale and other advantages seems a worthwhile venture. This is where the ActiveX rubber meets the road.

III. ACTIVEX BACKGROUND

Microsoft ActiveX controls are of concern to the pro audio community. This technology allows designers of computer-controlled sound systems to create common front-end software control panels that operate different manufacturers' units, without having to know anything about their internal code or protocols. This is powerful. If manufacturers incorporate ActiveX controls, systems designers will not be limited to products offered by a single, platform-specific manufacturer.

Each ActiveX control is made up of Properties and Events. ActiveX control Properties are values associated with the control which include things such as level settings, mute condition and meter readings. ActiveX control Events tell the computer something has happened, such as a switch closure, button press or clip detection. ActiveX allows the manufacturer to create an object (a piece of software code) which fully describes a device, while hiding the implementation details such as protocol from the programmer. By hiding the communication details, there is no need for manufacturers to agree on protocol. This lack of a protocol standard means that cooperation between manufacturers is not required. It allows each manufacturer to choose the best protocol for their devices.

For example, no longer would you need to know that the 17th byte of a 32-byte status message meant that the unit's second output channel was muted. With an ActiveX control, you might simply refer to the device's output 2 mute status as "Device1.Out2Mute."

What is ActiveX anyway?

ActiveX is a Microsoft-developed software technology released in 1996. ActiveX, formerly called OLE (Object Linking and Embedding), is based on the Component Object Model (COM), but provides substantially different services to developers. An ActiveX control is a unit of executable code, such as an .exe file, that follows the ActiveX specification for providing software objects. This technology allows programmers to assemble reusable software controls into applications and services. However, software development using ActiveX technology should not be confused with Object-Oriented Programming (OOP). OOP is concerned with creating objects, while ActiveX is concerned with making objects work together. Simply stated, ActiveX is a technology that lets a program – the ActiveX control – interact with other programs over a network, regardless of the language in which they were written. ActiveX controls can do similar things as Java, but they are quite different. Java is a programming language, while ActiveX controls can be written in any language (e.g., Visual Basic, C, C++, even Java). Also, ActiveX runs in a variety of applications, while Java and Javascript usually run only in Web browsers. ActiveX controls can be used in web pages and within visual programming languages such as Borland's Delphi, Sybase's PowerBuilder, Microsoft's Visual Basic and even in tools such as Adobe's GoLive and National Instrument's LabVIEW.

For pro audio applications, objects are the sliders, buttons, indicators and other graphical screen entities. The objects have properties: slider position, slider range, on or off for buttons and indicators. Once the screen objects are chosen and placed, ActiveX controls can link the objects' properties to other ActiveX controls such as the device parameters inside an audio device. For example, linking a slider to the ActiveX control for a device's level control. Then, moving the level control graphic slider varies the audio level and vice-versa.

Here's an example: A computer is used to control an audio system over an Ethernet network and something on the computer's screen controls some function of the system. The idea is to place controls on the computer screen and link them, using ActiveX, to a parameter in the system. What's important is that only the controls required by the computer's end-user need be displayed. Additionally, more detailed interfaces including hidden or password-protected web pages can then be created to provide any level of system parameter access desirable — from complete system control, to a lone system power button, or anything in-between. No longer are systems limited to the number of security levels provided by a vendor's software, nor are you limited to controlling a single system parameter per screen control. For example, you can link multiple ActiveX controls to a single screen object, thus adjusting EQ level simultaneously with master level control and limiter threshold. You can also program actions when certain events occur, such as triggering audio playback or turning a system off at a certain time or adjusting delay time as the temperature changes.

You can control different parameters inside the same device from different computers on the network as well as control the same parameter from multiple computers. This is one of the major advantages of networks – multiple control locations automatically update when changes are made by any control location.

However, ActiveX controls are not limited to just Ethernet implementations. You can create RS-232 or MIDI capable ActiveX controls. For audio networking purposes, Ethernet-enabled ActiveX controls are discussed here. It is wise as a development architecture to separate the communications code from the other software development pieces to allow portability of ActiveX controls onto other transport mechanisms.

One of many popular software packages used to create user interface web pages for computer-controlled systems is Microsoft's FrontPage 2000. Web pages may or may not be accessible over the Internet. Using FrontPage 2000 or any of the many ActiveX-ready software packages, ActiveX provides, literally, an infinite number of programming possibilities.

The procedure for using ActiveX with FrontPage 2000 is simple: insert a manufacturer-supplied ActiveX control in your web page. Set the control's Properties – such as the IP address and name for the manufacturer's device. As needed, insert a button (or scroll bar, check box…) and set its Properties like the button's name, value and control range. Use Microsoft's Script Wizard that is included with FrontPage 2000 to link the button's Events to the device's internal ActiveX parameter(s) or link other Events and Actions as needed for your application.

IV. NETWORKING RELAYS, SWITCHES AND INDICATORS

Many systems must incorporate relays, switches and luminous indicators for control, event or status indication. A product[1] has been developed that links on-screen ActiveX objects through a 10Base-T Ethernet network to the device's logic ports. The device's logic ports are capable of driving relays or LEDs, or reading switches, relays or any zero to 5 volt source. Thus, on-screen objects can be linked with these real world hardware tools to incorporate their use in networked systems. Applications for the zero to five volt input port include switches, but temperature or humidity sensors, potentiometers or any variable 5 volt source are suitable. Since this is a networked system, multiple locations can control and monitor the logic ports, and the on-screen status always remains linked to the hardware tools – change either and their states always correspond. This product is one way of implementing relays, et cetera into network controlled systems.

V. CONTROLLING EXISTING EQUIPMENT

Using existing RS-232-based and other serially-controlled equipment on a network is a likely requirement as the audio industry transitions into the network world. It may be some time before Ethernet control jacks are found on a wide variety pro audio equipment. Until then, carrying RS-232 data over an Ethernet network can be achieved in several ways. Several vendors sell inexpensive RS-232-to-Ethernet adapters. The previously mentioned Rane product simultaneously supports both the logic ports and the proprietary RW 232 protocol of Rane products. Also, many CobraNet devices offer asynchronous transport of RS-232 data. You are still faced with RS-232's one-to-one Master-Slave issues with these devices. Some CobraNet devices now incorporate memories that are recallable from contact closures. These memory closures can also be transported over the network allowing a single switch to recall memories in multiple, similarly-equipped CobraNet devices.

Logic ports on existing devices can drive or be driven from relays that are monitored by the network. Or, combining existing ports with those on the previously mentioned Rane device's logic ports, one can easily incorporate non-networked products into networks. Applications for these input and output logic ports include memory changes and/or system monitoring.

VI. ACTIVEX EXPERIENCES

ActiveX control of system parameters provides several advantages:

World Unity

ActiveX unifies divergent protocols from multiple manufacturers without the need for cooperation. ActiveX can significantly widen product choices for system designers and end-users, providing economies of scale, cost, availability, flexibility and serviceability.

Manufacturers need only develop ActiveX implementations for parameters requiring network control. Though, easier said than done, it's a small price given the advantages and direction of the pro audio industry. For some applications and products, ActiveX may eliminate the need for a manufacturer to implement a customized software user interface for a product or products.

Customization and Unlimited Security Levels

ActiveX permits easy customization of user interfaces and access levels, thereby avoiding constraints to the security levels offered by individual products or manufacturers. Complete interface customization, specific

1 Rane Corporation Via 10 Ethernet Bridge. The Via 10 product ships with ActiveX controls which allow its logic port states to be linked to a web page ActiveX control.

for the system end-user or the installer can easily be incorporated. The use of full color graphics with photos of the system controls or its components can be placed and scaled on a computer screen. The common password-protected web pages and firewalls of the computer world offer the security necessary for any system.

Third Party Software Support & Availability

ActiveX is supported by many software vendors who offer third-party education and support avenues. Sources of information abound for those who want to dive into this technology. Development of ActiveX controls is possible using any of the previously mentioned tools. With a variety of ActiveX capable packages available, needs from the simple to the advanced are satisfied.

Microsoft's Visual C++ includes a library called Active Template Library (ATL) which is designed to allow programmers to avoid the inefficiencies of other less efficient approaches to ActiveX. While ATL does allow development of ActiveX files that can be more size and time efficient, the drawback to this approach is the added difficulty in debugging the code.

Same Business Model as AMX & Crestron

ActiveX offers the same business model to the audio industry as room controller software providers. Many sound contractors either have in-house programmers for such software development or this service is contracted out. Additionally, these same sound contractors may already have a web master on staff that is capable of ActiveX implementations for sound system control.

Those who have followed the trends in room controller technology already recognize the industry's acceptance of Ethernet and IP-based implementations. Thus, it is a simple step for room control developers to support ActiveX controls. Plus, their development tools are already web-publishing based, so current room control developers should have a small learning curve in the ActiveX world.

Therefore, for network systems with one or more PCs as the main controller, instead of assigning Panja (formerly AMX) or Crestron programmers the system control programming tasks, you find a web site provider who can implement the system control front-ends.

The success or failure of this new relationship remains the same: the system's control implementation relies on the system designer's ability to communicate the needs to the software provider. This is no different than the current room controller business model.

Non-proprietary and Ubiquitous

ActiveX controls are widespread and non-proprietary. Web page creators have been utilizing ActiveX controls for years, thus providing a large and knowledgeable group of ActiveX-familiar providers. ActiveX also provides the ability for these providers to create their own customized ActiveX controls. The color, apparent texture, size, shape, shading, look and feel of each control can be created completely from scratch and designed to match any on-screen décor imaginable.

Fits well into existing PCs

ActiveX is easily incorporated into existing PCs running Windows and Internet Explorer avoiding the need to create a new technology, protocol or hardware-based solution.

Multiple, Simultaneous Control Locations

While not an ActiveX advantage, the combination of network technologies and protocols permits incorporation of multiple simultaneous control locations. This was a previously difficult task in the control

industry because serial streams such as RS-232 and RS-485 are unable to share multiple masters. With web-based architectures like ActiveX, controls on multiple web pages offer simultaneous control of the same system parameter(s) from many locations. For the CobraNet ActiveX controls implemented, Simple Network Management Protocol (SNMP) was used to provide the control capabilities found in computer network management, control and diagnostics tools such as Hewlett-Packard's OpenView. The case where changes made from a device's front panel are reflected in multiple network control screens is also satisfied. This is one of the major advantages of networks – multiple control locations can automatically be updated when changes are made by any control location.

Another advantage to network technology is its inherent redundancy ability. With careful network design, both the control and audio transport paths offer automatic fault redundancy.

ActiveX controls also have a few limitations, the first of which is easily overcome in CobraNet systems once the SNMP capabilities of CobraNet mature.

Signal Indication

Perhaps the most important need within controlled systems is the need for accurate and trustworthy on-screen signal indicators. ActiveX control signal indicators have yet to be implemented within a CobraNet system. The current CobraNet implementation for such signal indication is inadequate. A new implementation of CobraNet's SNMP capabilities should solve this problem.

Timing

ActiveX has timing limitations, particularly as the number of controls on a single web page are increased. As controls are added to a page, more RAM is used and the multiplexed processing time allotted for each control is diminished thus creating practical timing limits. The response times are not easily measured or calculated. Timing is a function of the computer's speed, the network's speed which can be constantly changing, the ActiveX control's size and code efficiencies and – in the case of CobraNet-based systems – how often the CobraNet interface and the CobraNet device itself scans for network and/or hardware changes. In the CobraNet case, the ActiveX control data is being transported over SNMP.

While it is impractical to suggest precise delay times for system changes, a unicast CobraNet system with 120 ActiveX controls on a single page, utilizing four, 24-port managed switches, with 64 audio channels and two simultaneously running web pages appeared to have a response time around one second. This is not heart warming, but the intent is to communicate performance levels to allow applicability decisions.

Time-critical applications such as show control and synchronized events may not be deterministic enough for ActiveX.

Uncommon and New to the Audio Industry

Few pro audio manufacturers offer ActiveX implementations of their software-controlled parameters which creates a Herculean obstacle for significant implementations across multiple manufacturers' products. Also, there is no current support for ActiveX in Netscape or the Macintosh world.

There are certainly other ways to implement cross-manufacturer control. For CobraNet-based systems, the forthcoming CobraCAD software from Peak Audio offers significant advantages through its use of SNMP – particularly as the CobraCAD package matures. No other currently available tools seem to offer the advantages of ActiveX for the pro audio industry.

VII. CONCLUSION

ActiveX provides a viable solution to the problem of user interface across disparate equipment in computer-controlled systems – particularly CobraNet and other Ethernet-based systems. While no solution – including ActiveX – can solve all problems, ActiveX appears to be a viable one at this juncture for the above issues.

References

[1] Proceedings of the AES 13th International Conference on Computer-Controlled Sound Systems (Audio Engineering Society, NY, 1994).

Bibliography

Tanenbaum, Andrew S. *"Computer Networks"* 3rd edition (Prentice Hall, ISBN 0-13-349945-6, New Jersey, 1996).

Perkins, David & Evan McGinnis, *"Understanding SNMP MIBs"* (Prentice Hall, ISBN 0-13-437708-7, Upper Saddle River, 1997).

Bates, Jonathan, *"Creating Lightweight Components with ATL"* (Sams, ISBN 0-672-31535-1, Indianapolis, 1999).

Cook, Devin, *"Emerging Standards for Networked Audio System Control,"* RaneNote 144, 1998.