Modern Approaches to Control Systems Design

An InfoComm International® White Paper
# Table of Contents

**Acknowledgments** ........................................................................................................... 1

**Executive Summary** ......................................................................................................... 2

**A Modern AV Control System** .......................................................................................... 3

  - The User Layer ............................................................................................................. 5
  - The User Interface Layer ............................................................................................. 5
    - Interface Hardware ............................................................................................... 6
    - Graphics Layer ...................................................................................................... 6
    - Navigational Layer .............................................................................................. 6
  - The Platform Layer ...................................................................................................... 7
  - The Enterprise Layer .................................................................................................. 8
  - Signal and Data Transport ....................................................................................... 8

**The User** .......................................................................................................................... 9

  - Who Are the Users? .................................................................................................. 9
  - Discovering User Requirements ............................................................................. 9
  - Defining Functional Requirements for the User Interface .................................. 10
    - Navigational Design .......................................................................................... 10
    - Functional Narrative ......................................................................................... 12
  - Defining the Control System Scope of Work ...................................................... 12
  - Using the Functional Requirements ...................................................................... 13
  - Automating a Modern Control System .................................................................. 13

**The User Interface** ........................................................................................................ 15

  - Interface Technologies .......................................................................................... 15
  - User Input and Output Devices ............................................................................ 15
Table of Contents

Mobile Device-Driven Control ........................................................................................................... 16
User Interface Design .......................................................................................................................... 18
Navigational Design ............................................................................................................................ 19
Graphical Design ............................................................................................................................... 20
Modern User Interface Elements ....................................................................................................... 20

Control System Platforms ............................................................................................................... 23
Platform Components ......................................................................................................................... 23
Proprietary vs. Open Platforms .......................................................................................................... 24
System Types, Applications and Functions ....................................................................................... 24
Programming ....................................................................................................................................... 26
Software Scope of Work ..................................................................................................................... 27
Software Testing and Performance Verification ................................................................................. 27

Integrating With Enterprise Systems ................................................................................................ 29
The One-Touch Meeting ...................................................................................................................... 29
Middleware ......................................................................................................................................... 30
Data Processing and Analytics ........................................................................................................... 30
Communicating With Building Management Systems ...................................................................... 31

The Glue: Signal and Data Transport ................................................................................................ 32
Physical vs. Logical Network ............................................................................................................... 32
Communications Protocols ................................................................................................................. 34
Security and Authentication .................................................................................................................. 34
Bandwidth ............................................................................................................................................ 35
Wireless and BYOD ............................................................................................................................. 35
Cloud/Data Center Connectivity .......................................................................................................... 35

Conclusion ........................................................................................................................................... 37

References .......................................................................................................................................... 38
Acknowledgments

InfoComm International would like to thank the following industry experts for their generous contribution of time and content:

Timothy Albright, CTS®, DMC-E, EAVA, Director of Operations, Innovad
David Barnett, CTS-D, Senior Consultant, The Sextant Group
Timothy W. Cape, CTS-D, Principal Consultant, Technitect
Steve Greenblatt, CTS, Control Concepts
Derek Joncas, Manager of Product Marketing — Control Products, Extron Electronics
Jean Lee, MS Applied Cognition and Neuroscience, HCI, Independent User Experience Consultant
Brian Morris, CTS, Product Manager, Kramer Electronics
Howard Nunes, CTS, President and CEO, PepperDash Technology Corp.
John Pavlik, Director, Engineering Architecture & Design, Crestron Electronics
Rem Remington, Tower Access Coordinator, Crista Media
Shaun Robinson, VP Product Management, AMX US
David Silberstein, Director, Technical and Sales Education, Crestron Electronics
Executive Summary

Modern control systems design begins with a deep understanding of a set of business challenges. In the past, design thinking centered simply on people having a meeting or conducting a class. But today, audiovisual solutions are highly contextualized. AV systems have become business tools, and the people who use and manage those tools are driven by two dynamic forces: AV/IT convergence and the proliferation of smart devices.

We now operate in a post-convergence industry. AV-enabled rooms are no longer stand-alone or one-off systems. They are interconnected components of larger systems, which must be managed and supported, and they are strategic investments, which must be analyzed and justified. The context of AV is evolving. AV is a strategic enterprise asset and it must be planned, funded, implemented and managed as such. Bandwidth, topology, security, intersystem operability and other strategic concepts, such as total cost of ownership (TCO) and return on investment (ROI), are the new language of AV. “Converged AV/IT” really is “enterprise AV.”

In the context of enterprise AV, the industry has been forced to redefine its concept of the AV user as well. The industry has to think beyond people in a room — or even people connected to a room remotely. In an enterprise, AV is part of an ecosystem of stakeholders whose needs and demands must be considered when designing control systems. AV professionals must consider who will be monitoring, supporting, analyzing utilization and forecasting facilities requirements for a solution. What are their needs? How can AV professionals support their needs in the design of the hardware, software and integrated systems? How can they factor in network administration, help desk and support ticketing, systems maintenance, and integration with third-party systems, such as building management, access control, scheduling and enterprise planning?

Too often, solution providers assume they already understand the business challenges facing their clients and they apply a known solution without first asking important questions. It’s tempting to use cookie-cutter templates for control systems design because it is easier or cheaper. And that is not necessarily bad for the client. But without a context-sensitive analysis, solution providers cannot know — and should not presume to know — their client’s needs. In reality, the typical AV system of just a few years ago is obsolete by today’s expectations, and those expectations are constantly evolving.

Workplace environments are constantly changing as well — sometimes chasing fads while at other times pursuing the latest trends in productivity and health research. Either way, AV professionals owe it to their clients to evaluate their current and foreseeable needs carefully.

This document explores the ways that industry professionals today approach AV control systems in the context of enterprise AV and mobile, smart devices. The authors have done their best to identify noteworthy trends and differences from the recent past, and to provide practical guidance for identifying the client’s business challenges and overcoming them through a modern control system design.
A Modern AV Control System

Whether they do it from a tablet computer, traditional touchscreen or button panel, today’s users have to be able to accomplish whatever their AV systems were designed to accomplish without having to worry about the control system. Moreover, it doesn’t matter how users will interface with a control system if the control system hasn’t been designed in a structured, thoughtful way. A control systems design model should address all the various pieces that make up a control system — from the users themselves, to the interface, to the hardware, software and network connectivity — and do so in a way that supports users’ needs and preferences.

An AV control system takes into account users’ need to collaborate with other users and to control their environments in a simple and seamless way. And it supports their ability to control that environment using everyday office tools or the latest consumer- and professional-grade technologies, including mobile devices, smart-building automation or cloud technologies.

What goes into today’s enterprise AV control system? It is certainly more than AV. By using a control systems layer model, AV designers and integrators can describe clearly the interconnected people, processes and technologies that factor into control systems design. The control systems layer model shows how various components relate and interact — not only with each other, but also with the user. Through the model, AV professionals can ensure the finished system works as required and supports whatever way users prefer to interface with their AV.

What follows is an overview of the control systems layer model, followed by in-depth discussions of the various layers as they pertain to modern control systems.
A Modern AV Control System

Control Systems Layer Model (courtesy of Technitect and PepperDash)
The User Layer

As its name implies, the **user layer** represents the various users and stakeholders of an AV control system. More importantly, it addresses their needs and requirements. (See The User section in this document for a deeper discussion of this layer.) User needs drive the functionality and configuration of a control system, as well as the overall AV system. Once an AV system is implemented, end users interact with the system via an interface or through third-party applications, such as scheduling software.

Functional requirements define the tasks that a control system must perform based on input from the user interface or related systems. In this context, automation logic is not actual software code. It is a description of what the software needs to do. Automation logic works in tandem with navigational logic in the user interface to create a coherent picture of how a system works and how it will deliver the right user experience.

One way to document automation logic is through a button-by-button and action-by-action description of what user interface elements do when triggered. Such a description should also contain information about user and enterprise tasks the system will perform, which may be related to inputs other than the user interface (e.g., from a scheduling system). Developers use this information as a guide in programming the system.

The User Interface Layer

The **user interface layer** defines how the user directly interacts with a system. (See The User Interface section in this document for a deeper discussion of this layer.)
The user interface controls or obtains information about the underlying system(s) being controlled. It is what the user sees and interacts with to perform tasks. The user interface can be separated into three distinct layers: **interface hardware, graphics layer** and **navigational layer**.

### Interface Hardware

The user interface always includes some type of hardware, which can be separated into two categories: visual and non-visual. Visual interface hardware is what a user sees (and usually touches) to control an AV system. Non-visual interface hardware includes devices that do not include a direct visual component. These include near-field communication (NFC) devices and many types of sensors, which gather information about the user (e.g., their identity, location or physical actions) and the environment (e.g., temperature, humidity and light). Control systems can use such information to execute commands and initiate AV functions or pass the information back to the user via the visual user interface or audio feedback.

### Graphics Layer

The graphics layer is what the user sees on the visual interface hardware, including colors, fonts, lights, buttons, graphs and images. On a touchscreen or web page, it is the pixels that make up what the user sees. The graphics layer is the skin over the navigational layer.

### Navigational Layer

The navigational layer is the skeleton beneath the graphics layer. It consists of the buttons, text and layouts over which a graphical skin is laid.

The navigational layer defines parameters, such as which and how many buttons appear on a control page or surface, their general layout, where and when images or windows should
appear, and when page flips or other changes in what a user sees should occur. All the information on the navigational layer is independent of the graphics layer, but does not include information about colors, fonts and backgrounds.

The navigational logic relates to the user interface itself. For example, if a user presses a button or swipes the screen, then a designated change in the user interface occurs. Logic that defines what non-interface devices in a system should do — or takes input from other systems — resides in the platform layer (see below).

From a design perspective, first the navigational layer determines how an interface should function, and then the graphics layer is applied.

The Platform Layer

The platform layer is where the brains of a system reside, in the form of software. The software performs tasks defined in the layers above. It may also issue commands to other devices or systems, either to perform certain actions or gather information. (See Control System Platforms for a deeper discussion of this layer.)

The platform layer includes software, hardware and cloud-based services. It may include proprietary software running on proprietary hardware, offered by various control system manufacturers, or non-proprietary software running on open systems, such as web-based solutions. In any case, the software performs tasks based on the automation and navigation logic defined and documented in the preceding layers.

The hardware component of the platform layer is the muscle of the system. It includes the hardware that runs control system software, such as control system processors and mobile devices, as well as the hardware that is controlled by the software, such as projectors, switchers and digital signal processing devices.

The cloud component of the platform layer includes systems that run on a different network nearby or on a hosted system that is accessed via the Internet. With cloud-based systems, their physical location is irrelevant and the software running on cloud-based systems is often made available as a subscription service.
The Enterprise Layer

The enterprise layer comprises other systems that run on an enterprise data network. Such systems usually require some sort of application program interface (API) — part of the platform level — to exchange information. And transport protocols process the intersystem communications data and transmit it to and from enterprise systems. (See Integrating With Enterprise Systems section in this document for a deeper discussion of this layer.)

Signal and Data Transport

Almost every hardware device and software program requires some form of signal and data transport to connect and exchange information with other hardware and/or systems. (See The Glue: Signal and Data Transport section in this document for a deeper discussion of this layer.)

<table>
<thead>
<tr>
<th>Physical Infrastructure:</th>
<th>Signal and Data Protocol:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet, Serial, Wi-Fi, Bluetooth, IR</td>
<td>TCP/IP, UDP, RS-232, RS-485</td>
</tr>
</tbody>
</table>

The physical infrastructure is the collection of wired and wireless devices and cabling that make up the pathways from one device or system to another. This includes switchers, routers, base stations, antennas and other signal and data transport hardware, including the cabling and patch bays that connect them.

To carry data and signals over the physical infrastructure, data and signal protocols are required to allow devices to communicate data between them. This includes many protocols and standards that vary depending on the devices and systems that are communicating with one another and the physical infrastructure that connects them.
The User

Technology systems are conceived, designed, implemented and maintained to address a user, business or environmental need. A system design should start with users to determine not only what they need the system to do, but also the larger organizational context in which they operate. To create a solution that best facilitates user and organizational goals, control system designers need information about the cultural and technological environment in which the system will be implemented.

This section describes the process of gathering requirements from control system users. It then covers how integrators may automate a control system based on those requirements.

Who Are the Users?

Users may include a range of stakeholders who interact with, invest in, budget for, technically support, and benefit from a control system. They have a vested interest in its successful functioning. Control system users may be:

- Presenters, instructors, meeting participants and students — both local and remote
- C-level leadership (CEOs, CIOs, CFOs, CTOs and COOs)
- AV and IT technical support staff
- Instructional and/or media designers
- Communications staff
- Facilities staff
- Budgeting, accounting, contracting and purchasing staff

Determining the users in an organization can be challenging — as challenging as determining their requirements.

Discovering User Requirements

When it comes to enterprise AV control systems — large or small — defining the scope of work starts with determining the organization’s mission. What type of business is it? What is the culture like? What are the politics? Designers and integrators should meet with users and stakeholders to get a complete picture of their general and specific needs. Focus groups designed to determine user experience may include the following questions:

- Who are you?
- What do you do?
- What does the person sitting next to you do?
The User

- How do you interact with others?
- What tools do you need to have that kind of interaction and collaboration?
- What do you see as problems in the way you use technology today?
- What do you want to do with technology?

Defining Functional Requirements for the User Interface

An AV system’s functionality is tied closely to the control system interface. The user interface can be simple, such as a handful of hardware buttons on a wall panel, or more elaborate, such as a high-definition, color touchscreen. It can include various modes of operation or unique and separate interface elements tailored to different types of users.

To develop today’s control system interface, designers/programmers gather functional requirements. They then design and implement the control system in the context of enterprise AV systems. The document that captures these elements, usually referred to as a functional specification, can take many forms, but it should include a few critical elements.

Navigational Design

The navigational design should include a wireframe layout of the visual user interface surface that depicts the buttons, interface objects and other information that will be displayed at any time. For multi-page interfaces, there should be at least one layout shown per page. The wireframe should be graphics-neutral and not representative of the colors, fonts and backgrounds that might ultimately be used. Think of it as a simple block diagram-like representation of the layout.
Example of a functional wireframe (courtesy of PepperDash)

The intent of the wireframe layout is to show what kind of interactive and informational elements may appear on each page, plus where and how they might be arranged. Once approved by the user, the graphical design can be applied to the wireframe to show its final form.
Functional Narrative

In conjunction with the wireframe layout, a functional narrative describes how the user interface should operate, defining what each button or object does or displays. Some buttons may only present a different page to the user. Others may trigger a command or series of commands to devices throughout the system, causing the state of those devices or their parameters to change. Objects may reflect the status of local or remote system parameters, whether those parameters were changed by a user or automatically.

Text (and perhaps diagrams) may also describe other non-visual functional system requirements, as associated with the visual user interface. For example, motion or light sensors may change the status of the system or trigger an action, such as closing shades, adjusting lights or simply shutting down the system because the space is unoccupied.

The narrative may also include text describing interaction between the AV system and other enterprise systems, such as lighting, building management and HVAC.

The functional narrative is crucial to determining accurately the level of effort required for a control system design.

Defining the Control System Scope of Work

When it comes to integrated systems, it can be difficult to separate the control system scope of work from that of the overall AV solution. Functional descriptions of the audio and video systems that support users’ needs will also define the functional requirements of the control system. In addition, there should be descriptions that identify the context and requirements of the control system, specifically at the enterprise level. Consider, for example, the following enterprise-level questions:

- Is there an AV monitoring system?
- Is logging data needed for equipment, system and room usage?
- Will the system need to interface with enterprise security, access, scheduling, building automation, energy-tracking or other systems?

AV designers and integrators must also take into account enterprise standards, equipment/vendor guidelines, user interface standards and other requirements that organizations employ more frequently to better manage technology resources in local, regional and global locations.

To create a comprehensive scope, designers and integrators should document several complementary elements of the functional requirements:

- The overall enterprise mission and the context for the AV and control systems
- Requirements associated with the enterprise environment
- The functional requirements of the audio, video and communication elements of the system(s)
- The user interface requirements
The documentation should include a narrative and schematic signal-flow diagrams for each of the types of systems to be implemented. These requirements play a key role in both the user interface and automation logic layers of control system development.

Using the Functional Requirements

Once the functional requirements have been established and documented, integrators use them throughout the process of developing and implementing not only the control system, but also the overall AV system.

First, functional requirements force consensus and agreement among stakeholders; the requirements set expectations for the remainder of the project. Gaining agreement is usually an iterative process, serving not only as a validation of user needs, but also as buy-in from the stakeholders.

Second, functional requirements act as a contractual guide for the AV provider’s team to design and develop the hardware, user interfaces and underlying programming. In many cases, the requirements are included in the document package for project bidding, so the more detailed it is, the cleaner the bid pricing will be.

Third, requirements are used for the systems performance verification guide at the end of the project, to compare what was installed and programmed with what the customer needed.

Thus, a vigorous and well-documented functional analysis should occur as far upstream in the process as possible. If it is omitted in the early stages of a project, it should be inserted as soon as practical. If a control system designer/developer must guess at functional requirements during system installation, it’s likely the control system will fall short of its intended purpose as a business tool and strategic investment.

To establish where the end user can achieve productivity savings, integrators perform a set of tests prior to defining and setting up automation. Such tests may involve installing occupancy sensors in the client’s conference rooms and allowing them to run for a set amount of time (e.g., three months). That helps to determine how many people attend meetings and how many times meetings really occur when they’re booked and, therefore, set the usage benchmarks that help to create more efficient meetings. Other measuring tools include setting up monitors to assess a room’s operation and determine the best equipment choices for the room. Once the integrator defines those parameters, automation can begin.

Automating a Modern Control System

The concept of automation is to cause multiple events in sequence based on a single input or trigger.

When users press “Videoconference,” for example, they should be presented with a dialing directory or keypad, but automation should activate cameras, displays, microphones, amplifiers, lights and everything else needed to start the video call. The order and timing of all these items is important for the overall user experience.
Most examples begin with an active input from a person (e.g., a button press on the touchscreen), but it is not the only case for automation. Sensors play a large role. If a meeting is scheduled but no one arrives in the room, for example, the occupancy sensor will never change its state. If the control system is monitoring both the schedule and the sensor, the system can wait 15 minutes and then turn off the room’s equipment and lights and remove the meeting from the calendar, so the room is available for others to book.

Equipment can also initiate automation events. A projector that needs its filters cleaned or changed can send an email asking for service. The astronomical clock can close the shades as the sun comes around to reduce cooling costs. Lights can be dimmed by 20 percent and projectors’ brightness lowered because the building management system has issued a command for systems to shed power load.

Regardless of the automation or series of events, it is critical that the feedback loop is completed. It is up to the control system to get confirmation of a change in status because this can change the action or timing for other events. As an example, directing a projector to turn on is not enough. What if the projector were just turned off and it needs a five-minute cool-down before it can turn back on? The control system should poll for status to verify that the projector is off before sending an “on” command, or verify that is on before sending a lamp status query.

Automation makes usage simpler, prevents user error and enables a smoother user experience.

It takes time to create all the possible automation paths for a control system, which can lead to a more expensive overall system. But a system that allows people to start or join a meeting in significantly less time has strong value in an enterprise environment. Looking at increased productivity over time, the owner can quickly recoup the automation investment and potentially show a profit.
The User Interface

Users’ expectations have been radically altered by the introduction of smartphones and tablets. Such devices have changed the way people interact with technology. AV designers and integrators must take into account the implication of new expectations when creating a control system user interface. Doing so requires thinking about technology in two related-but-separate ways: First is the way today’s users interact with technology; second is how AV pros can actually deliver the expected experience using technologies that are currently available. Approach from the perspective of the contemporary AV user — their worldview and mindset — designers and integrators must navigate between the psychology of the user interface and the mechanics for implementing that user interface, from a programmer’s or technician’s point of view.

These days, the user interface (UI) is a subset of a larger function — user experience (UX) — based on the way people interact with technology. When a user says of a control system, “I want it to be as intuitive as my smartphone,” he or she is talking as much about the UX as the UI. Or to put it more starkly, consider a UX without the UI, such as when a user says, “I don’t want to have to tell the AV in the room what to do. The room should know what I want to accomplish. It should be intuitive.”

Control system providers need to reimagine the conversations they have with users in order to understand their interpretations of “intuitive,” in the context of a user interface and a user experience.

After all, the control system interface often serves as the primary point of contact between the users of an AV system and the devices and applications that provide functionality. To be successful, the user interface must be responsive to users’ needs as determined by the functional requirements. The following section describes the components and techniques involved in creating user interfaces.

Interface Technologies

In addition to the traditional touchscreen or button panel, there are a variety of other devices and methods for interacting with a control system, including smartphones and tablets. The first major differentiating factor among interface technologies is whether an interface is an input or an output device. In many cases, a single device may function as both simultaneously.

User Input and Output Devices

The traditional AV control system input device is some sort of touchpanel or button panel/keypad. But a control system can use other devices (e.g., microphones and cameras) as inputs as well.
Increasingly, input devices have grown to encompass consumer electronics. Many AV manufacturers have written applications that allow people to use their smartphones or tablets to control their systems in some way. Every manufacturer has approached the process differently. Some manufacturers ask programmers to create an app that acts the same as a touchpanel. Others create apps that include features not found in a touchpanel or other interface, such as service controls (see Mobile Device-Driven Control, later in this section).

In addition, when it comes to current control systems, various types of sensors can trigger events without requiring a user to interact directly with an input device. Motion, occupancy, lighting and door-trigger sensors can all be part of an automation system.

Input devices can work quietly in the background or require active user intervention. Often, the same devices can operate in either mode, depending on application and preference. For instance, an occupancy sensor might be considered a passive input to a system because it is triggered automatically, whether or not a user is aware or desirous of the activity. A light switch might be an active input, because the user has to walk over to it and activate the switch in order to trigger the input.

User output devices may include displays, loudspeakers, lights, signage, haptics (i.e., any form of interaction involving touch) and smell output devices (i.e., devices used by restaurants and movie theaters to emit scent on demand). Any video signal (e.g., HDMI) can cause a system to turn on based on the signal presence. A switcher reports to the control system that there's been an event change. A microphone can sense a signal and report it to the control system. Beacons (e.g., Bluetooth, wireless network and RFID) are capable of sensing a mobile device, activating an interface and turning on a system. Many devices (e.g., smartphones, tablets and wearables) may serve both as inputs and outputs.

Mobile Device-Driven Control

It today’s enterprise world, where workers are influenced by technology trends in their personal lives, software developers may need to plan for multiple levels of user complexity. They may need to provide user interfaces for touchpanels, browsers and/or mobile devices. In an important meeting, for example, a presenter may use a table-top touchscreen, while a technician may have a tablet with duplicate or additional controls for adjusting AV settings as the meeting goes on. Meanwhile, a support technician may be monitoring remotely using a browser-based version of the room touchpanel.

Control systems manufacturers support mobile device app development to some degree. Such apps may be created at the same time as touchpanel graphics. In some cases, pages in a touchpanel can be saved simply as apps; in others, the app version may need to be quite different. When adapting control system interfaces to mobile devices, developers have to address issues such as orientation and rotation, and they should fine-tune apps for specific mobile device operating systems or environments. But often, the bulk of work in creating a mobile experience for a control systems interface has already been addressed with creation of touchpanel graphics.

Mobile platform manufacturers limit what runs on their devices. They also provide curation and distribution requirements that can be burdensome for app developers. For this reason, control system vendors normally provide a shell program and manage the versioning, curation and distribution themselves. Control software developers create files that can easily be loaded into
those shell programs. In this way, they provide control system user interfaces on mobile devices. Whether the mobile app user interface mimics the control touchpanel, or provides greater or lesser capability, is a design decision that should be explored in the user requirements gathering and documentation process.

Mobile Device Challenges

The central challenge with incorporating most smartphones, smart watches and tablets (collectively, smart devices) into enterprise AV systems is that, with a few exceptions, they are fundamentally retail consumer devices. They do not have the level of centralized control and management that most IT departments prefer. Operating system updates can happen unexpectedly, which may cause control manufacturer shell apps — and the custom user interfaces that run within them — to fail or become unreliable. Control system manufacturers must contend with a number of different operating system versions and device requirements that each brings its own set of application development challenges related to screen size, processing power and network security.

Enterprises that choose to rely on mobile devices as their primary user interface run the risk of having a room disabled for a period of time.

There are also a number of physical challenges to incorporating such devices into an enterprise environment. For example, providing continuous USB power to a consumer device (such as a tablet) that’s designed to run on battery power can be difficult. Mounting such a device, running USB power cables — which present distance limitations — or locating “wall warts” (AC-DC power converters) are all issues that must be addressed when a customer requests a mobile device as a control system interface. Power over Ethernet (PoE) converters may help solve some of these challenges.

Wireless signal management is another issue. Consumer devices tend not to include robust wireless management features. In a corporate environment, where wireless security is critical, ordinary wireless management techniques can interrupt the device’s signals or cause the device to go offline.

Another risk when incorporating mobile devices to control AV/IT systems is theft, which requires vigorous asset management. The IT department’s goal is to reduce risk, loss, and time required to maintain and support systems. And managing mobile devices in an enterprise environment contributes to all of those issues.

In short, the concept of walking into and controlling a room from a mobile device is a vision that remains difficult to realize. The geolocation component has to work so well that the smart device can identify the room and tap into the room’s control system seamlessly and quickly. Mobile devices support limited development environments for incorporating sensors and indoor geolocation. Furthermore, the operating systems of mobile devices can handle only a limited number of control system interfaces, which is a problem in larger facilities.

Lessons From the Residential World

Residential systems are increasingly influencing commercial systems design, based on advancements in home automation sensors, thermostats and wirelessly controllable appliances. And as users enjoy these control experiences in their home, they will expect similar in their work environments. Many of these technologies are important to a smart building. As
organizations use control systems to better manage their AV systems, the two areas of control will continue to intersect.

In residential environments, control apps on smart devices are more common. Residents expect to be able to turn on lights, adjust thermostats, disarm security systems or start home appliances from a smartphone. Other technology capabilities have also changed the way people utilize their residence. Remote monitoring and control, for example, support the concept of aging in place — allowing elderly persons to maintain their independence longer. People who travel can monitor their homes and respond to visitors at the front door using the intercom system even when they’re far away.

Wearables (i.e., smart watches, glasses) have also begun to play an important role in consumers’ lives, which means manufacturers must account for another framework in their app development. The programming rules for wearables will differ from those for smartphones, mostly because they’re an extension of the smartphone. Screen size and interface response are also different, and the number of control functions that may be placed on a wearable is limited.

In short, today’s AV control systems are becoming part of the Internet of Things (IoT), an all-encompassing consumer electronics/enterprise networking concept in which all devices include an Internet connection. In the consumer electronics world, we’re beginning to see so-called hubs from a variety of manufacturers. Hubs can control more than one IoT component at a given time and will have implications for modern control systems. The hub concept has not yet been fully automated, but it enables users to control more than one device within a single application, for example, a Nest thermostat and a Nest smoke detector.

With IoT, users should be able to access systems — whether they’re control systems, AV systems or sensors — using a computer, smartphone or other Internet Protocol (IP)-enabled device. The interface to do so may be an app or a web page, resulting in myriad user experiences that must be accounted for by control system designers and integrators.

User Interface Design

Usability of an interface design is essential to the success of your projects. Investment in the design of interfaces directly affects the customer’s ability to understand and use the products created for them. Although closely aligned with purely visual graphic, audio and textual design, interaction design provides an overall enveloping focus on how well and naturally users interact with technology.

There are many ways users can interact with a controllable system. Some options are non-visual, such as motion sensors and NFC technologies, and some are visually oriented, such as touchscreens and button panels.

Non-visual sensors that provide status information to the control system regarding identity, motion, occupancy, vacancy, temperature, light, weight/presence and other environmental variables can be incorporated into the user interface in different ways. Such elements may affect the user’s control and navigation of a system, provide status indicators for users and support staff, and send out information to support other administrative functions, such as facility support, asset management, scheduling and security.
On the visual side, touchscreen interfaces typically have multiple pages with sub-pages and other navigational elements. Digital signage may include a touchscreen overlay that allows users to interact with content, in which case the user-controllable elements must be considered in concert with the content being displayed.

Hardware panels are usually much less dynamic, but may still include small touchscreens or other variety of video or LCD text display feedback. Hardware panels without screens may be considered a single-page user interface with little or no feedback available to the user on the panel itself. Hardware buttons may have printed labeling, and may or may not provide feedback in the form or lighted buttons or panel-embedded indicator lights.

Regardless of the type of user interface, AV designers and integrators typically create a navigational design upon which to apply a graphic design.

Navigational Design

Navigational design dictates the quantity and general layout/location of user interface elements. These elements include controls (i.e., buttons, sliders, and directional pads) and can be combined with other visual navigational tools, such as 2D and 3D gestural inputs, depending on the user interface hardware being used.

Navigational design serves a purpose greater than decoration; it is an important tool for effective communication. The organization of information on a screen can be the difference between a message users understand and one that leaves users feeling overwhelmed. Even the best interface can suffer and go underutilized if the visual presentation does not communicate well. It is important to understand how users function in the real world and to interpret this behavior correctly for user interfaces. For example, users read a screen in the same way they read other forms of information — left to right and top to bottom. Moreover, the eye is always attracted to colored elements before black-and-white elements, isolated elements before elements in a group, and graphics before text. Also, to develop user interfaces for the widest audience, the navigational design should focus on the most common user.

Navigational design incorporates the following key elements:

- **Gestures and movement** — Gestural navigation is the most common interaction method utilized across mobile platforms today. This form of navigation differs from traditional touch navigation in that the interface recognizes finger movement after contact and responds accordingly. Think of a person using a finger to rotate a graphic or spreading two fingers to zoom in on a part of the screen. Multitouch navigation requires a minimum of 1 cm (28 pixels) between any two fingers for the gesture to be recognized as two independent finger actions. Special consideration should be taken when designing for two-finger swipe to ensure the interface accommodates enough space for the user to touch with both fingers and complete the gesture while still allowing for the minimum spacing requirement.

- **Motion design** — Motion design defines how things move on the screen and is a critical part of an interactive experience. Motion design should never be gratuitous; animation always supports content and the experience as a whole. Transitions are used to provide critical clues and to make sense of application states. They also allow the user to understand where tools have gone when they are in a hidden state.
Graphical Design

Although navigational design establishes defining behaviors, gestures and responses, it is visual design that brings those elements to life. The key to successful graphical design is a design that subconsciously teaches the user. The user can visually see where to touch or slide without explicit instruction. This is challenging because, although visual design should add beauty and branding to the experience, it should never distract from the content. Designing for successful interaction means creating an extensible design vocabulary with specific attributes — a language of shapes, forms, colors and controls that visually guide users through tasks to meet their goals. A well-designed user interface is built on principles and a development process that centers on users and their tasks.

Graphical design incorporates the following key elements:

- **Icons** — Icons are pictorial representations of objects. Most current interfaces utilize a standard icon set that should be repurposed for iterations of your interfaces. Icon states are the result of an action taken. Once an action is taken on an icon, the icon reflects that action by showing its state.

- **Buttons** — Buttons are the direct interface the user has to the system. They should not be so small the user can’t use them. After that has been determined, the size of the buttons is assessed in relation to the size of the screen. It should be proportionally large enough to be seen and used — but not too big and ungainly.

- **Typography** — Fonts should be agreed upon by both the client and the designer. Depending on the color scheme and size of the buttons, some fonts simply should not be used. Again, this is a discussion with end users. It comes down to the look and feel they are trying to accomplish in the user interface.

- **Color** — Color is very important in the visual interface. It can be used to identify elements in the interface to draw the user’s attention (e.g., the current selection). Color also has an associative quality; users assume there is a relationship between items of the same color. Color also carries with it emotional or psychological qualities; for example, a color can be categorized as cool or warm. However, when color is used indiscriminately, it can have a negative or distracting effect. Misuse of color can cause an unfavorable user reaction and can hinder productivity by making it difficult for users to focus on a task.

- **Animation** — Animation can illustrate the operation of a particular tool or reflect a particular state. It can also be used to include an element of fun in the interface. Designers can use animation effects for objects within a window and interface elements, such as icons and buttons.

Modern User Interface Elements

Beyond the foundational elements, there are several recent factors that should be considered for graphical interface design.

**Feedback**

In the context of the user interface, feedback occurs when a touchpanel displays information about the state of controlled devices. Feedback may be delivered through buttons, text and/or
graphics. It can be as simple as power status, projector lamp hours or the current volume level. Or it can be fairly sophisticated, such as the type of signal passing through a transmitter or the video resolution of a scaler. This information may be important for users to understand if they're interacting with the system correctly, or for technical users who must troubleshoot or diagnose issues.

Notifications

Notifications are alerts that may need to be brought to the attention of people in a room and/or required by law. Notifications may include fire, weather, campus-lockdown or other emergency alerts. Notifications may be incorporated into touchpanels, in-room displays or digital signage as pop-up windows. They are usually accompanied by an audio tone or flashing lights. A signal from an emergency notification, fire or life-safety system triggers such notifications.

The text and/or icons that need to be displayed may be mandated by law or negotiated with the owner. Typically, an emergency notification will appear over whatever else is being displayed and users may not disable or close notices until the condition has been cleared by proper personnel. The control system programmer should consult with the owner representative, security or emergency team, and the architect on how the notifications should be cleared once an emergency condition has passed.

Energy Management

Today’s AV control system is expected to play an important role in energy management efforts, whether it’s to actively minimize energy usage or simply inform users of their consumption. It’s important to understand that the level of energy management may depend on the trade-offs the users are willing to accept. For example, it’s possible to maximize energy conservation by programming a control system to power off all the AV equipment unless a meeting is taking place. However, such an approach could cause meeting delays when the equipment must power on again, which can take time. Often, touchpanel-configurable schedules are useful to power off equipment overnight and partially power up equipment during work hours (i.e., standby mode). More sophisticated approaches use occupancy sensors and meeting calendars to actively manage equipment power and, as a result, balance conservation and convenience.

The ability to switch between “green” (more conservation) and “non-green” (more convenience) modes may be available via a toggle on the user interface itself. Green mode may reduce touchpanel or display brightness levels, or shorten the amount of time before the touchpanel dims or devices go into standby mode.

When it comes to displaying energy consumption information, a best practice is to display such device data as analog gauges or charts, or as dynamic text. Colorful or animated graphics present information that is easy to understand, as well as aesthetically pleasing. However, for more granularity or detail, text may be more appropriate. Often a combination of the two is the best approach.

In a smart-building environment, where the AV control system is integrated with broader building management systems, the AV user interface can provide a secondary point of control for those systems, and the AV control system can incorporate smart-building automation logic. For example, the AV control touchpanel may include a page for room temperature, lighting or shade management. Or it may integrate an equipment rack’s energy management schedule.
into the HVAC system for temperature or humidity control. AV control systems can also be used for daylight harvesting when integrated with shades and lighting systems.

**Accessibility and Code Compliance**

Some locations or jurisdictions require specific user interface enhancements to enable persons with disabilities to access and use AV systems more conveniently. These may range from the physical location of a touchpanel, to font sizes or color schemes, to auditory or haptic feedback related to button presses. Designers, integrators and programmers should check local and national laws, as well as the owner’s own requirements governing accessibility. Then they should verify that their specific interface is compliant.

Auditory feedback is usually the literal reading of a button as it's pressed. For example, pressing “Computer Input” could trigger an audio file stating, “Computer Input has been selected.” Haptic feedback is available on some touchpanel interfaces. It creates a touch sensory feedback (vibration) when a button is pressed. Programmers may also need to offer larger font formatting for button labels and other text, as well as color-neutral themes for visually-impaired users. Accessibility options can be made available as a toggled option on the user interface.
Control System Platforms

A control platform is made up of computing and communication hardware, an operating system and software tools, which allow the intended system functionality, user activities and operational instructions of the AV system to be carried out. Though the required actions may be platform neutral, the method of their implementation varies from platform to platform.

Some platforms are basic, structured and offer limited capability, whereas others are more expansive, have fewer boundaries and provide more power. The range of platforms varies by cost, ease of development or degree of required training. This section explains platform components and choices, as well as key programming concepts.

Platform Components

Most AV control platforms are based on specific or proprietary hardware that includes a microprocessor and communication ports to carry out instructions defined through software tools. Additionally, many platforms have proprietary user interface devices, such as keypads, handheld remote controls or touchpanels, which provide input to the system. A platform typically contains a programming tool, software language or configuration wizard that allows programmers to build an instruction set, which may be referred to as the program source code. Different platforms use instruction sets or source code in different, proprietary formats. Source code written for one platform is typically unusable in other platforms.

It is likely that a separate software tool is available to design the user interface for a touchpanel or other graphical user interface device. A separate file — different from the programming source code — defines the functional and graphical elements, as well as navigation logic, required to operate the control system.

Program source code and user interface source files usually also have a separate component, called the compiled file or object code, which is either loaded into the control system processor or the user interface hardware. These files are not editable.

In addition to source code and compiled code, a program is likely to have other components that are required to complete this package. These would include drivers or modules that provide protocols necessary for the operation of specific devices, or configuration files used to define variables within specific systems. Such drivers and modules may be available from device manufacturers; otherwise, the system programmer has to create them. All associated files are necessary to maintain and upgrade a system.

Connected to a control platform, device hardware may come in different configurations and satisfy different needs. Most will employ the common control protocols: relay/contact closure, digital I/O, infrared (IR), RS-232, RS-422, RS-485 and IP. The control platform selected for a given integrated system must have communication ports sufficient to communicate with all devices connected to it, which is normally verified by the system designer.
Proprietary vs. Open Platforms

Until a few years ago, all control platforms were proprietary, meaning that the manufacturers did not share information about their systems, and their own tools and components were needed to use the platform. Recently, control systems have begun using commonly available operating systems, which, along with flexible software and powerful processors, have opened up an opportunity for less costly and more versatile control systems.

The following are some examples of platforms seen in the industry today:

- Traditional AV control systems
- Android-based computers and mobile devices (Android apps running on Android devices)
- iOS-based mobile devices (iOS apps running on an Apple device)
- Open-source computers (e.g., a Linux-based computer running executable program written in C# or other industry standard language)

System Types, Applications and Functions

Although integrated AV systems may have various types of equipment and come in different configurations and capabilities, they can be categorized by both their physical layout and the functions they perform. The type of system is defined by the functional requirements it is intended to serve.

There is a strong trend at the enterprise level to normalize system types (also referred to as room types) throughout the enterprise, and to reduce the total number of types while still meeting core functional requirements. The reasons for this are straightforward:

- Reduce the effort and cost associated with implementing and maintaining a system
- Create a uniform, appealing and simple user experience

Manufacturers have attempted to stay ahead of this trend by offering all-in-one systems, simpler programming tools and systems that can be used right out of the box.

Control systems are made up of one or more system functions. Each system function offers one or more variations on the operation. The specific operation depends on the needs of the user, the specified AV equipment and the chosen control method. Basic system functions include:

- **Presentation** — Selection of a source to be viewed on a display device with accompanying audio. Variations on presentation would include single-display presentation, dual-display presentation, display windowing, multiple sources and matrix switching.
- **Audioconferencing** — Controls for placing and receiving analog or VoIP phone calls and adjusting incoming volume levels. Variations on audioconferencing include direct dialing, preset store-and-recall and phone book integration.
- **Videoconferencing** — Controls for placing and receiving video calls, positioning cameras and adjusting volume levels. Variations on videoconferencing may depend on the type or brand of video codec used in the system. These variations include on-screen dialing, direct
dialing, preset store-and-recall and address book integration. Camera controls may vary based on the number or location of cameras and the camera interface. Camera direction presets are often used for ease of accessibility and can be initiated by selection from the user interface, push-to-talk microphone or microphone voice sensing.

- **Recording** — Controls for archiving audio and video from meetings or events.
- **Environmental controls** — Controls for selecting and possibly modifying preset lighting scenes, controls for raising and lowering window shades manually or by selecting a position preset, and controls for viewing and changing HVAC set points. Lighting and shades may be controlled directly or automatically triggered by selecting a presentation source or initiating a videoconference.

The following are typical system types commonly employed in commercial or educational environments:

- **Classroom** — A system used by an instructor for presentation to a small or medium-sized audience, in flat-floored or smaller tiered rooms. Classrooms typically have a single-flat panel display or projector. Depending on the size of the room, voice reinforcement may or may not be provided. Operation of the system should be straightforward and consistent, providing only the basic control required for the instructor to operate the system.

- **Lecture hall** — A larger version of a classroom that is typically used for presentation to a medium or large-sized audience, usually with tiered floors. Lecture halls often have multiple displays and may support audio- or videoconferencing functionality for distance learning or other applications. Due to the size of the space and number of participants, voice reinforcement is typically provided. Operation of the system should be straightforward and consistent for an instructor. Occasionally, there may be a need to perform more complex functions, so advanced controls may be provided on a separate area from the typical user controls, often referred to as a technician’s page.

- **Auditorium** — A large lecture hall that may have a control room for operation, in addition to operation from a lectern. Auditoriums may have one large projector or multiple display devices capable of displaying multiple sources simultaneously. The operator of an auditorium may be the presenter, working from a simplified user interface at a lectern. Or, it might be a technician working in a control room, using a more advanced user interface that provides flexibility in operation and access to system adjustments that would not be appropriate for a presenter.

- **Performance space** — A unique type of system that is used for live performances and may also have a secondary application as a lecture hall or auditorium. Performance spaces typically have a control room and backstage controls as well as a portable lectern for lecture hall or auditorium operation. Performance spaces typically do not use a display for live performances, but often have a large projector. An advanced technical user who needs functionality rather than simplicity typically uses the primary functions. Multiple points of control with different user interfaces may be needed. The operator for lecture hall or auditorium operation would be a presenter or technical operator, and the controls would be similar to those types of spaces.

- **Entertainment system** — A unique type of system that is typically seen in restaurants, bars, museums and other public gathering areas. This type of system may be used by a non-technical operator or a technician to control power of displays, select video sources to be viewed on one or multiple displays, and select audio sources to be heard in one or more
areas. These systems may include presets for quick access of commonly used functions (e.g., recall of cable TV channels, routing configurations or volume levels). Additionally, the use of time-based functions may be implemented to schedule routine functions that occur on a daily or weekly basis as well as sequenced events, such as background music or digital signage.

- **Huddle room** — A small room or nook with a simple system, used for small-group collaboration, usually containing a flat panel display and a wired or wireless laptop computer interface. Operation of the system is intended to be straightforward with limited controls that are easy to operate by the participants. Huddle spaces may or may not include a physical user interface. Such spaces may simply be operated by sensing occupancy or the presence of a video signal.

- **Meeting room** — Larger than a huddle room, this type of system would be operated by any of the meeting attendees and may include multiple computer inputs, media players, and audio- or videoconferencing capabilities. The operation of a meeting room should be straightforward, simple and reliable, so anyone can comfortably operate the system with little or no assistance.

- **Boardroom or executive conference room** — A special type of space that is reserved for meetings of executives or dignitaries. Boardrooms typically have higher quality finishes and equipment, and advanced functionality. They are often operated by a dedicated user or technician in order to accommodate changing needs and provide a high-end, “white-glove” experience. Separate user interfaces may be included for the non-technical executive who needs quick access to basic operation with little confusion, and the technical operator who needs flexibility and control over the entire system. A boardroom may also have special features such as microphone-activated camera presets.

- **Multipurpose room** — A special type of meeting room that is configurable for multiple applications or meeting types, often including multiple rooms that can be operated independently or combined in various configurations as well as one large space. Operation of a multipurpose room should be easy for a non-technical user, while providing advanced technical controls for a system operator. Since a multipurpose room is by nature a flexible space, control typically includes the ability to reconfigure the room, combine rooms, make system adjustments and support users. Many considerations must be factored into the operation of the space when it is reconfigured for various applications.

**Programming**

In the world of AV control, the term “programming” is used to describe a process in which a software developer (often referred to as the programmer) uses the system’s functional requirements to develop, install and test the user interfaces and automation software required to operate the control system appropriately. A well-defined and detailed set of functional requirements is critical to the overall success of a project. The software scope of work can be considered a subset of the overall requirements, from the software developer’s frame of reference.
Software Scope of Work

The **software scope of work** will be of great value and a guide for the software development team. However, this guide may not have enough detail or direction to develop software that meets the expectations of the design team and the customer. The functional requirements will have a narrative of system goals, but programmers need much more detail before they can begin their work.

Interface design teams often supply layout and button-by-button descriptions. However, it is not just the input from a user interface that creates system automation. For example, the system may and should have a room occupancy sensor that turns the lights off when the room is not in use. But what about the projector and other AV components? Should they be turned off? If the functional requirements call for connecting to room-monitoring software, what items should be monitored and why? These items are not normally defined as part of the user interface.

Although it often seems easier for a programmer to rely on personal opinion or previous experience and not a detailed software scope of work, two problems may occur as a result of such an approach. First, the programmer may feel strongly about how a system should work, but other team members may have a dramatically different vision or understanding. Second, and most critical, is that without a complete software scope of work, the software development team cannot create a complete and valid testing and verification procedure.

From the programming point of view, if the software scope of work is complete and the software tests and verifies correctly, a return visit or update will not be needed. With today’s complex systems, however, it is very difficult to test for all possible combinations based solely on the developer’s imagination. Without a complete functional requirements specification and software scope of work, the testing procedures are going to be flawed, potential errors will be missed, and the system will likely fail.

Software Testing and Performance Verification

Testing is hard work and takes time and preparation. This needs to be factored into the initial project planning. Programmers have to think about how to test before they have written the code.

As devices become more complicated, the interactions between them become more intricate. If even a small system is examined closely, thousands of potential code paths could be found, many not even visible to the programmer. Software developers and integrators cannot — and should not — be required to test every potential automation path in a system. Such an undertaking would be exceedingly difficult and pointless. But the team should test every business case represented in the functional requirements and verify that the system will perform as intended when used as intended.

System Testing

Software testing begins, in a sense, in the user interface design stage, when the graphic designer or software developer steps the design team thorough the wireframes. All business cases or usage scenarios should be discussed and verified to be correctly represented in the proposed user interface.
In some instances, the design team creates “dummy” user interfaces to verify all requirements and explore alternative navigation paths. In other instances, they may create pilot systems. These systems are very similar to production systems but are provided to stakeholders to validate theoretical concepts explored in the design stage or verify system requirements prior to going into production. Similarly, production systems are often staged at an integrator’s fabrication facility, where equipment is racked and software is run through preliminary testing prior to installation at the permanent site.

**Automated Testing**

Many environments have unit-testing frameworks, which help to automate the testing of many system components. Programmers write test code to verify if a given component is working as designed. During software development, for example, the programmer may create test code to stimulate the system in an accelerated manner. If a user might press a button on a touchscreen once per meeting, for example, that button press can be simulated every five or ten minutes to test weeks of activity in only a few hours. Automated testing can also be done on a completed system.

There are a number of professional packages to automate an end-to-end testing cycle. Most often, they include some sort of macro-recording capability, which is run on a laptop or mobile device to simulate end-user behavior. Such test programs then provide either an output file comparison or a screenshot comparison. Because this sort of testing requires a baseline, this is what is known as regression testing. Regression testing provides a method to ensure that the software is acting as it did before. Manual testing is usually performed until the software meets its expectations, and then regression tests can be automated to allow the developer to concentrate on new features without concern about the impact these changes might have on the rest of the system.

**Diagnostics**

Diagnostics are also a big part of system verification. Control system manufacturers provide debugging tools in their development environments. These tools can be used to step through the code and identify where errors occur, or to provide logs that can be reviewed to locate errors after the fact. These tools might be used as a part of an automated test to compare previous results, or it might provide a pathway to diagnose an issue that is not easy to reproduce. Reproducing an error is often the most difficult aspect of diagnosing or troubleshooting a system fault. Often, there could be many potential causes of a symptom, and ruling out those causes can be very challenging.
Integrating With Enterprise Systems

Historically, AV control systems were only expected to control the AV system in a particular room. In most cases, aspirations were not even that lofty, and the control system often functioned as a glorified remote control, merely operating the devices in the system without adding much functionality or benefit.

Control systems are expected to perform much more than just room-based control. In order to achieve this, the AV control system is expected to interface with a plethora of other systems, including:

- Access control
- Asset management
- AV system monitoring
- Building management systems
- Cloud hosting and control
- Emergency notification
- Energy management
- Environment controls
- Events, catering, security and other services
- Helpdesk systems
- Scheduling systems
- Shared occupancy sensors
- Signage

The One-Touch Meeting

Today’s AV system is connected to other systems with which people interact on a daily basis. When users schedule a meeting in a calendar, such as Outlook, they should be able to set up the meeting and invite people, plus invite the room and have the room be ready when the user arrives.

The one-touch meeting is a major trend in AV, in which the only thing required of meeting organizers is to self-identify and acknowledge that they are ready to begin. One-touch meetings require more technical skill to enable and more effort from the design and implementation team. In a more complex system like that, the team might need .NET programmers writing the middleware code to enable the control system talk to a third-party API. Back-end integration may be needed with network credentials, RFID tags or badging.
Integrating With Enterprise Systems

systems, as well. Ideally, the AV system presents a single question to the arriving attendees: “Are you ready to start?”

Middleware

When control systems can be programmed using traditional languages like C#, IT professionals can add and maintain enterprise integration with a system and not have to learn yet another language. Configuration and room data can be exchanged and stored in an enterprise SQL server, allowing easy and consistent access globally.

Because C# can expose standards-based communication methods, enterprises can create systems using their existing skills in languages such as PHP and Python to interact with conference rooms. For example, by tying into the rest of the enterprise’s infrastructure, it’s possible to use badge scans to automatically turn on rooms and connect to meetings based on meeting reservations. It’s also possible to prevent individuals who didn’t book the room from turning on equipment.

Manufacturers also provide software development kits (SDKs) for .NET and Flash that programmers can use to communicate with their systems in third-party applications.

Flexibility is the key, because every organization has different needs.

Data Processing and Analytics

Analytics are normal and expected in an enterprise environment. AV/IT teams need to keep track of assets they’ve deployed. Which are being used and by whom? Which can be reallocated? What needs to be expanded? These questions can best be answered with analytics, but only if the underlying data is being collected from the local AV systems, aggregated and made available as reports. AV system analytics starts with the local control system monitoring all available information about itself, such as:

- Device states (Is the device on or off? What input or output is selected? What signal type is being sent or received?)
- Connectivity status (i.e., among devices, to the network, and to the server, last connection report, time connection achieved or lost)
- Usage time (i.e., time powered on and off)
- User login (i.e., individual identity, group association, log in time, log out time)
- Room occupancy (i.e., occupied, unoccupied, time of occupancy, density of occupancy)

By collecting very granular data, correlations can be made between the various data, such as which types of users engage which types of systems, when and how often, using which components and for how long, and whether rooms are being used for something other than their intended purpose. A very detailed picture can be extrapolated and used by the data analyst for many purposes. The most common purposes include:

- Device life-cycle tracking (i.e., average up-time before failure, which devices fail and at what rate, inventory forecasting)
MODERN APPROACHES TO CONTROL SYSTEMS DESIGN

- Next-generation design guidelines based on actual usage (What types of systems are most popular? Are there enough of them? Can less popular system types be repurposed?)

- Validating work environment decisions (Do huddle rooms support an open work space? Do people prefer desktop video versus meeting room videoconferencing?)

- Identifying negative behavioral trends (Are people booking rooms and never arriving? Are rooms booked for longer than necessary? Are recurring meetings left on the calendar long after people stop attending? Are people booking expensive videoconferencing rooms but not using the codec?)

- Determining return on investment (Have the systems made people more productive? Are people collaborating more effectively? Have the systems reduced travel costs? Have the systems lowered energy consumption? Are people using the technology that they requested?)

Communicating With Building Management Systems

Building management systems (BMS) are gaining popularity. And although the primary mission of most control systems is to provide in-room control, there’s a growing need to take whatever data is generated in the local environment and make that available to enterprise systems.

Some building management systems are very manufacturer-driven and are usually outside the scope of AV, so there may be a parallel process of control systems development that occurs on the BMS side via other vendors. The ways AV and BMS connect and the types of information they exchange could be very different, depending on the capabilities of the two systems. In a single-manufacturer environment, it may be very easy to get information back and forth between components. But between components from different manufacturers — or worse, between completely different categories of systems — intersystem communication can be very complex.

Communication with the environmental control can’t exist without a physical network or physical connection, protocols, some kind of hierarchy of permissions and a mission. It’s important to have a sense of purpose — form follows function — that meets users’ goals before enabling intersystem communications. Just because devices can connect doesn’t mean they should. Does every conference room need to be able to connect to a building management system? Does every system need a component that allows it to be monitored by another solution? Does every room need an occupancy sensor to detect if a person has shown up for a meeting? These are the types of questions designers, integrators and users need to work out before they tackle the challenge of integrating AV control systems with building management systems.
The network is the backbone of an AV control system. Ten years ago, in order to control a projector, the commercial AV industry used the RS-232 protocol almost exclusively. This required a dedicated serial cable from the control system to the projector or other controlled device. In today’s world, AV integrators typically connect the control system and most controllable devices to an Ethernet network using standard universal cabling or Wi-Fi, saving considerable cable installation cost.

Device management, monitoring and control, including mobile device integration, all require network connections. Today, without the network, there is no control system. This section delineates the physical from the logical network and discusses various aspects of networked AV devices, including protocols, security, bandwidth, wireless devices and cloud-based systems.

**Physical vs. Logical Network**

There is always a physical component to the network — a physical way that devices are connected. Ethernet networks are designed for scalability and security. To manage traffic and bandwidth, and to prevent inappropriate connections between certain types of devices, systems or users, networks are segmented and managed. The overall segmentation architecture is often referred to as the network topology.

AV segments are often separated from the general corporate network on their own physical or logical subnets. Through this approach, AV devices can be located in one part of the network, for example, building management devices in another, and email communications in still another. All of them are separated logically, if not physically.

The following diagram depicts this type of separation, using a dual-network-card control processor.
An AV subnetwork or “subnet” (courtesy of Crestron)

The biggest piece of this puzzle is the virtual local area network (VLAN). With a VLAN, all participants use the same physical wire, but a set of rules prevents them from crossing the lines. This concept can be likened to having three lanes on a highway and a sign that says, “No trucks in the left lane.”

While network topology is normally the IT department’s responsibility, control system integrators and software developers need to be familiar with networking concepts to understand all of the pieces of the network and work within it. Though the AV components would be part of the logical network, an AV professional typically would not be building that network.

At one time, AV components in a network environment were a mystery to the IT department. In the post-convergence world, the IT department owns and operates the AV equipment — the IT staff members have become stakeholders themselves. The key to success is to have the IT staff involved in the design process from the beginning so they understand the goals the AV team is trying to achieve and set up their network accordingly.

Network configuration involves no programming on the AV side, as it is the IT staff members who set and maintain network rules. But the AV team must understand the network’s IP addressing scheme and will typically obtain IP address ranges from the IT department to use for all the AV and control hardware/software to be installed.

The rules are the same for larger applications. If a control system needs to communicate with a building management system, for example, in order to adjust temperature from a touchscreen in the room, the IT department has to create that routing rule to allow the two systems to connect.
Communications Protocols

Communication protocols are the means by which a control system talks with the devices it controls. The protocols can be thought of as dialects and words, with the dialect being the type of protocol and the words being commands that are sent and responses received. The words can be aggregated into sentences, known as strings.

Common AV communication protocols include IP, RS-232, RS-485, Serial I/O, and IR, as well as the newer consumer-oriented protocols such as Consumer Electronics Control (CEC) and Universal Plug and Play AV (UPnP AV). A diminishing number of manufacturers continue to use proprietary protocols within their own brand families. Advanced protocols include extensible markup language (XML) and hypertext markup language (HTML). Note that a dry contact closure (relay) is often used for binary control but is not really a protocol.

Most control protocols require particular types of cables and connectors, which in turn require specific ports on the control system and controlled devices. The industry trend is to convert complex device control to IP and provide only an Ethernet port, which can handle control over other communications (e.g., firmware updates).

The type of control protocol that a designer chooses is usually a compromise among simplicity, reliability and cost of implementation. RS-232, for example, is tried-and-true reliable but requires additional cabling, has restrictive transmission speed and relatively short maximum cable lengths. It also requires the infamous null modem pin reversal in specific situations, which has confused countless installers. IP, on the other hand, supports universal cabling, high transmission speeds and long cable lengths. IP also supports optical fiber cabling for high-bandwidth transmission.

Security and Authentication

Security issues and rules are determined by the IT department. In most cases, an AV device that needs to cross over a network segment must follow the same security rules applied to the other network traffic. Users must authenticate to get access to the network, and, in many environments, AV devices must also authenticate before they will be allowed access to the network.

Many of the devices built by AV companies do not require authentication to access them, which may cause them to fail a security evaluation in a highly secure environment. AV professionals may rely on the customer’s IT departments to manage device authentication and security, or employ their own staff to install and secure devices subject to IT review.

Some AV system designs take advantage of user authentication to access network content, for example, for presentations. Portable presets — a more advanced usage model — allow users to access their favorite system settings in any room on the network. Some solutions allow authentication via smart devices using Bluetooth, RFID or NFC protocols. AV professionals need to discuss authentication requirements with the IT department in advance and select equipment that will support those requirements.
MODERN APPROACHES TO CONTROL SYSTEMS DESIGN

Once AV devices are authenticated and have the permission to share the network, the next step is to encrypt the data when required. There are standard and custom encryption packages from various manufacturers, which should be coordinated with the IT team.

Bandwidth

AV integrators need to know the approximate bandwidth requirements for the devices they are adding to the network and what the added traffic on that network will do to the overall performance of the system. For every system, bandwidth testing would determine that a device or system can operate correctly. If there is a bandwidth issue, it needs to be immediately addressed.

When bandwidth is a problem, one of the more expensive positions the AV industry falls back to is building its own physical AV network. Other than in super-secure environments, where “air-gap networking” is required, this is not generally considered a viable solution. In today’s world, remote system monitoring, calendar-scheduling integration, wireless webpage presentation and control from mobile devices all require quality, integrated network management.

Wireless and BYOD

When it comes to wireless devices, another level of encryption and security applies. Employees are not the only bring-your-own-device (BYOD) users. Guests and salespeople visit organizations with the intent to present some kind of content. Typically, corporate environments supply guest access whereby a non-authenticated employee can use the Internet without accessing the company’s secure network.

Increasingly, the AV industry is asking to install devices that need access to both “trusted” and “untrusted” segments of an enterprise network. This is “abnormal” for an IT design and will typically trigger heavy scrutiny from the IT department about how product security features are designed and documented. Attaching a wireless access point to a secure network for in-room content sharing, for example, may result in a visit from the security team within minutes.

Cloud/Data Center Connectivity

There is a growing trend in AV to provide managed services that the customer licenses for a periodic fee rather than buying outright. Such services can include a range of capabilities, such as system monitoring, voice bridging or even videoconferencing. Depending on the manufacturer, these services are hosted on either a physical machine (a server) or a virtual machine (VM).

A VM could be hosted in an off-premises server farm with other customers’ VMs (i.e., public cloud) or a dedicated data center (i.e., private cloud). This affords the customer more redundancy, data backup and protection from a variety of dangers, such as electrical surges or extreme weather.
Cloud-based videoconferencing bridges are becoming more common and many AV dealers are seeking the elusive recurring-revenue model by offering such solutions. Cloud-based control system services, on the other hand, are a new paradigm, with significant room for adoption and growth.

The goal of cloud-based control for audio, video, voice and data, lighting, security, digital signage, building management systems, shades, and HVAC systems is to enable remote, global management. Some manufacturers now offer programming-in-the-cloud solutions that even allow remote system configuration. In most cases, the local control system communicates to the cloud-based system for centralized management and data storage, and the local control system manages the in-room devices.
Conclusion

The shift to enterprise AV has taken the AV industry by storm, but it has occurred in parallel with another profound development — a social evolution that has changed users forever. In less than a decade, the mobile device, whether it’s a tablet or smartphone, has emerged as a universal and ubiquitous content platform. Users today expect a perfect, personal experience at all times. In the enterprise, network security professionals have to contend with consumer devices on their telecommunications and data networks. And the AV/IT industry as a whole will have to face an irreversible drive toward universal data access, full-time connectivity and intuitive automation.

It’s easy to envision a future in which users will be able to interact with a service or device in that space without actually having to touch or toggle anything. A small percentage of deployed systems already include this kind of technology, but in the future, this type of interaction will become ubiquitous.

We hope this resource provided you with some insight into the transitional steps occurring in the design of modern control systems, leading us to a future of pervasive AV use, with the simplest, most intuitive user interfaces possible — even no user interface at all.


