



## WHITE PAPER: THE END OF AV CONTROL COMPLEXITY

David H. Parish, Ph.D.  
President, Calypso Control Systems  
July 2005

Control systems are fundamental to an increasing number of Pro AV installations. With an absence of standards among devices and manufacturers, and with the common requirement of flexible and intuitive user interfaces, control systems establish the point of integration among otherwise incompatible devices and typically become the “face” of the solution to the industry’s customers. When working properly, the control system is barely noticed. Yet when any aspect of the overall solution falters, regardless of where the point of failure resides, the control system is often assumed to be at fault, e.g., the call comes in that the “buttons don’t work anymore”. In other words, getting the control solution right is essential to meeting or exceeding customer expectations.

For the most part, that’s been easier said than done. It’s ingrained within the Pro AV industry that control system development is complex, time consuming and expensive, expectations most relevant when dealing with solutions that include a graphical user interface. Typically, the development of graphical interface controls requires training, certification and highly specialized programming skills. Accordingly, Pro AV dealers are segmented into (a) those who can afford control system specialists on staff and can therefore pursue projects large enough to absorb those expenses and (b) everyone else.

As long as control systems remained a specialized segment of the industry, there was really no need to change the *status quo*. Two rapidly evolving industry developments, however, are forcing a control system shake-up. First, as mentioned above, control systems are no longer “specialized”. Control solutions are now required in a growing percentage of projects across an expanding number of markets and applications. This means that the “everyone else” category of system integrators needs a good control solution to bring to their customers if they plan on staying in the AV business. Second, the size of Pro AV project budgets is shrinking, just as the number of Pro AV projects is growing. Classic market maturity is underway, bringing with it the expected “good-news, bad-news” scenario of more projects, but fewer dollars per project. This second trend affects larger system integrators even more so than smaller ones. Their problem is not about skills, but about how to make sufficient margin on small budget projects.

Which brings the discussion back to control solution complexity. Complexity drives time and money, which ultimately defines end-user costs and profit opportunity. In other words, reducing the complexity of the solution is the key to addressing the evolving market, in that less complexity (a) makes solutions more accessible to all dealers, (b) drives end-user costs down and (c) drives dealer profitability up, even for small projects. So how do we drive complexity out of control solutions without sacrificing functionality or reliability? It’s useful to begin with an understanding of where the complexity comes from in the first place.

## ROOTS OF CONTROL COMPLEXITY

Control solutions consist of two basic components – control logic and user interface logic. Control logic refers to “if-then-else” type of programming typically required to insure that the appropriate control command, or more often, sequence of commands, is triggered when a specific event (e.g., a button press) occurs. User interface logic is all about creating an environment that provides the user with an intuitive interface for triggering control events. It’s worth mentioning again that user interface logic is relatively straight forward when the interface is restricted to physical buttons (e.g., relay or serial), but can get complex when a custom interface is graphically rendered.

The real complexity comes from the fact that most control solutions require that developers blend control logic and user interface logic within a single program. That is, the same program developed to render the graphical interface (i.e., what the buttons look like, where they’re positioned on the screen, etc.) must also contain embedded control logic. As a result, experts at graphical interface development (e.g., web developers) are not sufficiently skilled, because they don’t have knowledge of control logic (e.g., serial protocols for controllers a router). Conversely, experts at control logic (e.g., Pro AV engineers) typically don’t have interface development programming skills. So what happens? A sub-industry of developers who combine the two skills, often within the context proprietary programming tools, has evolved.

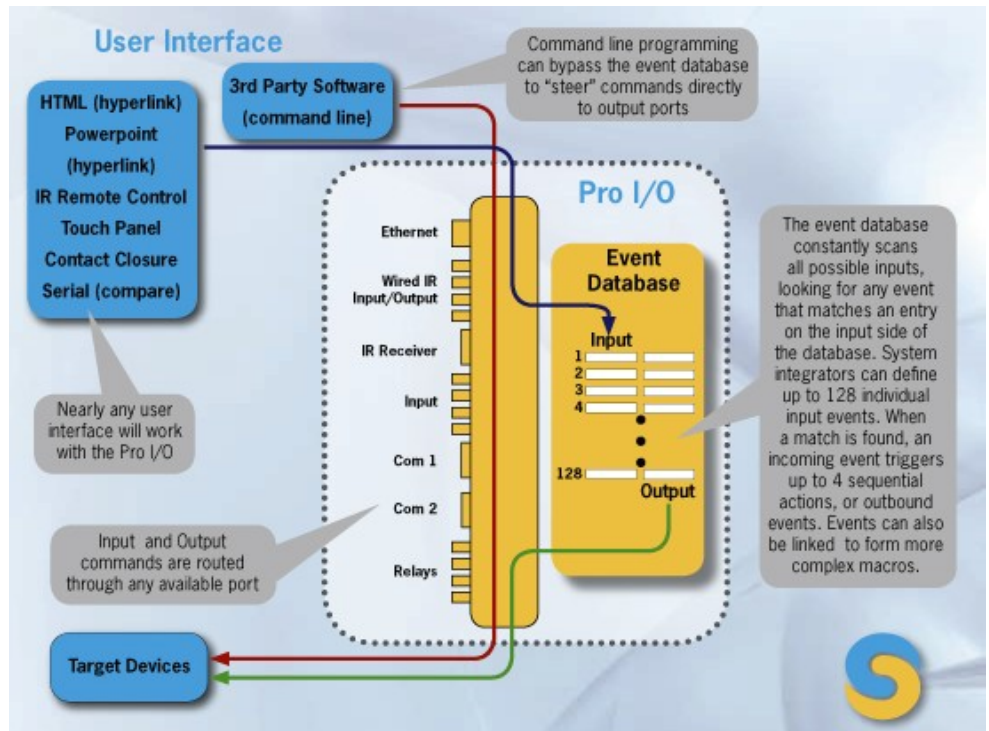
## SEPARATING CONTROL LOGIC AND USER INTERFACE LOGIC

If blending control and user interface logic is the root of complexity, then it’s clear that unwinding the two is the key to reducing complexity, driving down the cost of implementation and to making control solutions more accessible to dealers and end-users alike. This is precisely the philosophy behind Calypso Control Systems products, which are built around an open architecture approach for delivering control solutions to the Pro AV market.

### *The Event Database*

Calypso controllers, including the **Pro I/O**, are intelligent, network devices that offer a range of control functionality. Various models offer differing, but somewhat predictable, combinations of serial, IR, relay and Ethernet ports. That is, on the surface, one would see “just another controller” (though of course, as the inventors, we see a thing a beauty). Yet the Pro I/O separates from the pack in one very important way. Much of the processing power residing on the unit is devoted to managing the on-board, configurable “event database”. Simple in concept, but powerful in implementation, the event database reduces the process of defining control logic to defining incoming events and corresponding output events, or actions. An event is anything the controller is capable of receiving or sending, including IR, serial strings, Ethernet commands, and the state of inputs and relays.

Think of the event database as a translation table, capable of converting any incoming event into any outbound event sequence. In fact, within the Calypso architecture, each incoming event triggers up to 4 sequential, user-defined (actually, dealer-defined) actions. Again, an action is any serial, IR, relay, Ethernet or internal (e.g., trigger a new event) command. In short, the design supports “anything in and anything out”.



With this approach, for example, an IR signal sent from a \$10 CD player remote control can trigger a complex sequence of serial, IR, relay and Ethernet commands. Despite the flexibility of this approach, it is important to note that no programming logic (i.e., if-then-else) is involved. The process of configuring the event database simply requires knowledge of the customer’s workflow requirements, the list of equipment being controlled, and the control protocols for each piece of equipment on the list. In other words, it requires information that any competent AV integrator has to have. Nothing more.

### *Any Interface Will Work*

So if control logic is separated from interface logic and reduced to an event database that resides on the controller itself, what happens to user interface logic? The answer is ... anything you want. Because with an event database, user interface development is simply a matter of triggering the right event. The controller doesn’t care or know how an event is triggered. Rather, its job is to constantly scan its inputs, looking for something that matches an event in the database. As a result, any user interface tool or programming environment will work just fine, whether it’s a custom program, an inexpensive IR remote control, devices that open or close relays or sends serial commands or anything in between.

In other words, the developer of the control interface doesn’t need to know anything about “control logic”. They simply need to know which event to trigger in the database. This is, of course, the opposite of the current state of affairs in control interface development, which requires that the developer also be a control logic specialist. With

the Calypso approach, there is no barrier to having any web developer, for example, create a control interface, as long as the event database has been appropriately configured. Goodbye specialized programming, high complexity and high prices. Hello choices, simplicity and value.

## **GOING FROM GOOD TO GREAT**

The changes introduced by the Calypso Control approach to control system architecture are good – perhaps even very good. We’ve developed a fundamentally different way of creating and deploying Pro AV control solutions that delivers more value and choices to dealers and end-users, while reducing complexity. The great news is that this approach is not just a better path to the same result, but that it produces an entirely new set of benefits not attainable with existing systems. This is where the Calypso solution goes from being good, to being great.

### *Control HyperLink – c\_Link*

An important – perhaps the most important – benefit of the event database structure is that networked Calypso controllers are fully addressable through standard hyperlink commands. A hyperlink is the most common way for programs to communicate with one another, and is most often used to trigger a “jump” to a URL or web page, either within a program or by automatically connecting to or launching a second program. Click on a hyperlink within an email and you automatically launch a browser and jump to the right web page. For that matter, click on this hyperlink ([www.calypsocontrol.com](http://www.calypsocontrol.com)) and you’ll jump right from this document to our web site. I didn’t need to tell Word (what I used to write this white paper) or Adobe Acrobat (what I used to save and distribute this white paper) how to create the hyperlink. The programs simply recognized what it for what it was, and applied standard hyperlink logic. Hyperlinks are very simple, very effective and quite universal.

We use the term **c\_Link** to refer to Calypso’s patent-pending approach to triggering control events using hyperlinks. The goal of c\_Link is simply to trigger an event or sequence of events. Its implementation couldn’t be easier, as it requires only two pieces of information: (a) the IP address of the controller and (b) the number of the event to be triggered, as defined by the configuration of the event database (e.g., event 1 = lower screen, event 2 = projector on, etc.) that resides on the controller. A hyperlink, button or otherwise, containing these two pieces of information in the right syntax, will find the controller on the network, and trigger the event. From there, the control logic residing within the event database takes over.

Using c\_Link certainly makes a user interface programmer’s job easier, regardless of whether they’re developing in HTML, Flash, C++ or anything else. But that’s not the big news with c\_Link because with the event database structure, the programmer’s job already got much easier. The big news is that c\_Link makes it easy to embed controls directly within PowerPoint, Word and PDF files. The point to emphasize is that once the

event database is set up, no programming or control logic knowledge is required to build a control interface. For example, the syntax used to drop a c\_Link command in a PowerPoint button that triggers event number 45 on a Pro I/O with an IP address of 10.0.1.1 is <http://10.0.1.1/button.cgi?event=45>. That's it. Press that button within PowerPoint and event number 45 happens. The point here, of course, is that anyone can do it, given just a little bit of information. Moreover, there really are no limitations on how much control can be achieved, despite the simplicity of the approach.

Imagine a series of c\_Link buttons placed on a presenter's PowerPoint slides, providing direct and immediate control over projectors, lights, audio, video servers, DVD players, routers and anything else in the room. In fact, the devices being controlled don't even need to be in the room, as the PowerPoint event can trigger IP commands to devices anywhere on the network, including to other Pro I/O controllers. Imagine creating a c\_Link control interface using Word, saving it as a PDF file and then e-mailing it to a customer who can run it from their computer (Windows, Mac or Linux) or load it onto their handheld PDA. The thing is that hyperlink is built-in to all of these platforms and programs. And again, all the user needs is the IP address of the controller and a list of event numbers. Now that's open architecture!

### **CONNECT. CONFIGURE. CREATE.**

Let's face it. This changes everything when it comes to control. Just as the event database places control logic squarely back in the hands of AV technicians and engineers, c\_Link, removes interface development from the land of "specialty programming skills" and into the hands of standard programmers and anyone capable of using PowerPoint or Word. It's a great example of an open architecture design that exceeds the technical rhetoric and truly delivers tangible value. So go connect, configure and create with Calypso products. Chances are good you're already certified.