Essentials of End-User & System-Level Control Design & Programming for Presentation and Videoconference: Part 2 – *The Systems-Level*

Part 2-of-2: Version 1 - Submitted by S. R. Sharer – Communication Design Group Inc. – August 20, 2006
Part 1 of 2 was originally submitted in June / July of 2006 and is also available on-line.

Restatement of Part 1 - General Introduction: Many organizations are, at this time and for the anticipated future, deeply engaged in the deployment of high-level sophisticated integrated systems to be used for presentation and videoconference communications within large, medium, small-group and individual spaces, and for typical and atypical applications. These systems are being designed around many of the fundamental elements that are outlined and discussed in-depth as detailed in the full "**Control system Design for Video Communications / Videoconferencing"** document that was previously [initially] published in June / July 2006.

As a subset of all earlier documents - This document will not re-state all of the discussion points and design integration fundamentals of the previous (Part 1) document. **This "PART 2" document** (as an integral subset or follow-on chapter) **will continue to concentrate on the issues and elements of Control Systems *Design** & Control Systems *Programming*** (***Again** - these are two very **different aspects** of developing **any** Control System) **and the role these directly play in the matters of:**

- **managed input and display configuration** selections & choices, *and*
- **system & function redundancy** or **reliable-system(s) & function(s) - recovery** in the face of certain system or component failures.

(These two (2) areas will be alternately covered at random, not necessarily in the order listed here)

Ultimately, **this pair of documents** (***Part 1***, submitted previously, and this – ***Part 2***) along with other previously submitted / published documents, and a thorough understanding of the principles that are detailed for analysis and consideration, can serve as a guide for anyone responsible for mapping-out, specifying, integrating, configuring and programming the functional Distribution & Control Systems for visual presentation & communication systems for many different user communities.

Please Note: Especially in this "Part 2" chapter of "System-Level Control Design & Programming for Presentation and Videoconference": ***portions*** of the Control System functions we are about to discuss ***will be familiar*** to those experienced in this aspect of systems integration and design, and even though ***portions*** of our discussion ***will have "generally accepted and deployed" functionality commonly seen* in other integrated remote control systems applications, some of the integration and programming processes and functions detailed within this Part 2 document have no *generally and widely* known precedent** (at-least **none** that are **known to this writer**)* in any application or implementation that is now [or has been] in-use at end-user companies or within deployed specialty systems. ***This means that there will be some degree of exploration into configurations and programming flows that would appear to be almost totally unique***. The nature of this, along with the complexity of some of the operational elements we will outline for the Remote Control Systems, **may** often require time and additional monetary investment to develop, deploy and fully test.

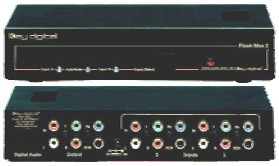
*Of course, we encourage others knowing differently to write, submit & share their own work on this broad subject matter of "**Design of Control Systems for Videoconference**" or on other topics.

Begin ***Part-2*** – **“System-Level Control Design & Programming”** We now return to our discussion of “Background Intelligence”, and, at this point, **we are going to step back from previous primary focus on the End-User** [though **not entirely discard** any consideration of them] **concentrating, instead, on ways we can use this Background Intelligence to directly influence and impact the performance reliability, and “‘fault-tolerance’ of”, the complete presentation and visual communication systems.** Through this approach we will, ideally, dramatically [and almost completely indirectly] impact or influence the end-user experiences and their general [and specific] feeling towards VTC technology, and any activities related to using the technology. **This is the area where we will begin to “chart new territory” for which there are, as previously stated in our introductory paragraphs, no *known* templates** (**again** - ***unknown to this author***, though it is certainly plausible that numerous templates are known to others who will, perhaps, begin to freely share them) from which a Control System Designer or a Control System Programmer can work.

In previous documents titled **“Redundancy: Hardening a Design, Integration or Communication Solution”** and **“End-User & System-Level Control Design & Programming [Part-1]”** we discussed an intricate interweave of review, planning and design for each area within the context of potentially atypical beneficial **automation of certain system functions.** We previously stated that **“certain circuitry”*could** [often should] **be added to the systems for automatic “cut-over” in the event of a failure within a particular component. Though not “unique”, this is not necessarily a common integration practice. *NOTE:** The ***Systems Designer*** **must determine which** of the **components** in any signal flow can and **should be bypassed through any** ‘**automatic cut-over**’ decisions made by the Control layer resident in a single unit in the flow, or the Control layer as resident within the Main Systems Control Frame**. In reference to the specific “automatic cut-over” we are mentioning here we mean:** the addition of individual electronic components that could automatically “self-trigger” a new activity-state, based on the presence or absence of a reference-signal, apart from any typical actions they might take resulting from end-user commands to a Master Control System. **There are numerous audio and video switches** *(and switches for signals such as Ethernet-based data, RF signals, digital telecommunication network signals, etc. that we do not have space or need to fully and completely list here in this article)*, **for instance, that will auto-sense certain specific signals and self-invoke a particular input-output configuration within themselves,** based on the state *(often – “presence” or “absence”)* of those auto-sensed signals.

Examples of simple / basic Auto-sensing Switches:

AB Components 4-In to 2-Out S-Video & Stereo Audio Autosensing Switch



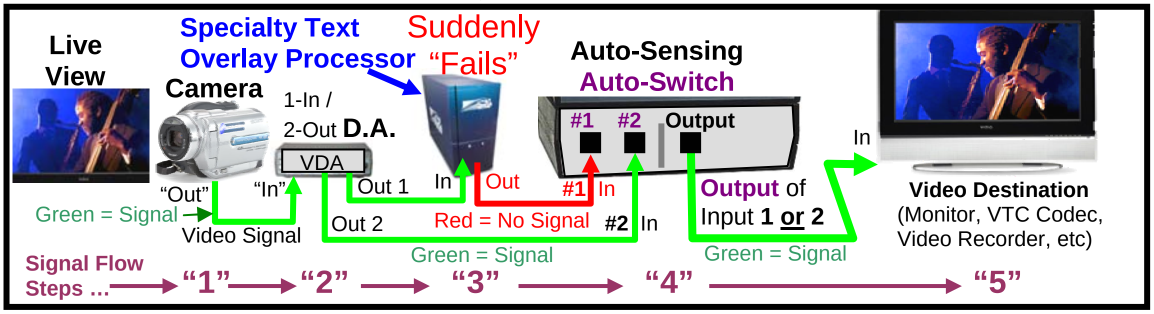
Key Digital 4-In to 2-Out Composite-Video & Stereo Audio Autosensing Switch

We also previously stated (in the “Redundancy” document referenced above) that, in order to remain cost effective, “cut-over” or “fail-over” for redundancy ***might*** be based on bypassing a failed unit, and that some element of overall lower / lessened system level-of-performance ***might*** be accepted, **if** a complete loss (bypass) of a single particular unit was not considered devastating to the communication (if the video communications were able to continue at a somewhat lesser level of performance or functionality). ***OK: Yes… Hold-on…*** there **are other choices**. In fact, **there are hundreds** (if not thousands) **of "choices" for approaching *many* difficulties, including this specific difficulty** of "maintaining source-signals" (data or carrier signals) without the end-users ever perceiving that there has been any "interruption" to the flow or fluid-exchange of information. ***REMEMBER: We are by no means at any time indicating*** that this design theory and resultant choices are "carved-in-stone" for every application out there, everywhere, at all times. **Nor are we stating here that** other designers or design specialists have not ever produced very good alternative methods for handling this specific problem of "what to do about retaining or maintaining source video/ audio/ network signals if an intermediate signal processing device suddenly 'fails' and goes off-line, consequently 'stopping' the signals at that point, preventing original source signals from reaching the required destination(s)". **There are, indeed, countless mechanisms** and approaches to the broad issues of "fault-tolerance" and "redundancy", ***but*** **- in the interest of keeping our discussion focused and as brief as possible, this initial choice we are discussing here is based on** *some sort of* **auto-bypass** *setup / configuration. Other discussions and other designers may provide other means and mechanisms…*

When selected as the option to use, this "auto-bypass" approach can [often] be the result of design for signal-flows that combines devices like those seen above with other differently-"specialized" units, using the auto-switches to maintain the base signals by invoking an automatic bypass of a particular "specialized" unit that might suddenly fail. **For instance,** this would be the approach option that had been selected if an auto-sensing switch was integrated into a designed system flow in order to "invoke" the switch as a result of a source-video signal that now failed-to-appear at an intermediate or destination processor point on a feed from a specific specialized source-video processor. The resultant "switch" or automatic "cut-over" would result in sending the original-source directly to, for instance, a destination monitor or encoder while bypassing the [now sensed as "failed"] specialized intermediate mid-stream source-video-processor device. ***To the end user,*** there ***might*** only be a slight "blink" in the video image as fed-to a video display unit or as fed-to and transmitted-by a videoconference or streaming encoder, but the source-image would [**normally**] return immediately after such a "blink", **as opposed to** [alternately, if no automatic "cut-over" was designed into the system flow] any images permanently "blanking" as a result of a signal feed attempting to flow through a now-failed component. **The only difference, often not realized by the end-user,** is that there ***might*** be some functionally favorable processing that [as a result of the "switch"] can no longer be done to the original source video* signal (*the camera signal in the graphic panel below), since the 'failed' specialized source-video processor would have been effectively "eliminated" or bypassed from the video signal flow. ***In the very simple illustration below*** we will attempt to graphically and then textually explain the precise nature and structure of this "flow" and "alternate flow" approach.

PLEASE: **CAREFULLY** REVIEW EACH **GRAPHIC** ***AND*** THE **DESCRIPTIVE TEXT EXPLANATION BELOW / BENEATH THE GRAPHIC PANEL**(S).

For example: (*remember – we are providing this only as **an example**, **not THE example** for all approaches and all actions at all times, everywhere, in all designs related to this over-arching topic*)**:**



In this example [*above*] , the "Specialty Text Overlay Processor" (3) *[for the purpose of this explanation, let's say that this processor provides the ability to display a momentary electronic text label along the lower border / edge of the video for the purpose of providing location or other subject information to the viewer of the video]* **suddenly goes "off-line / fails",** meaning that, **if** we had **not** planned a solution to this problem, then we'd lose the **Camera** image **(1)** as fed from the **Camera** - to the **Video Distribution Amplifier (2)** - to the **Specialty Text Overlay Processor (3)** and then out - to the **Auto-Switch (4)** – to the **Video Destination (5). The video would "stop" at failed-step-"3"** and never make it to steps "**4**" and "**5**".

As a result of our design setup where we **have** [apparently] determined that **this Specialty Text Overlay Processor is not essential to continue the communication** (we have previously determined that ***we can and will continue to use the video*** as supplied from the camera ***even if no text-label is overlaid*** onto the image before the image is sent to the Display or to an Encoder for transmission), **in this case** the **Auto-Switch auto-senses the loss of this Primary (#1) Input from the Specialty Text Overlay Processor and instantly switches to the Secondary (#2) Input** (the Camera image as fed from the **Camera (1)** direct - to the **Video DA (2)** and directly out - to the **Auto-Switch (4)** – to the **Video Destination (5)**). We can still "see" the camera image, **minus** any text overlay.

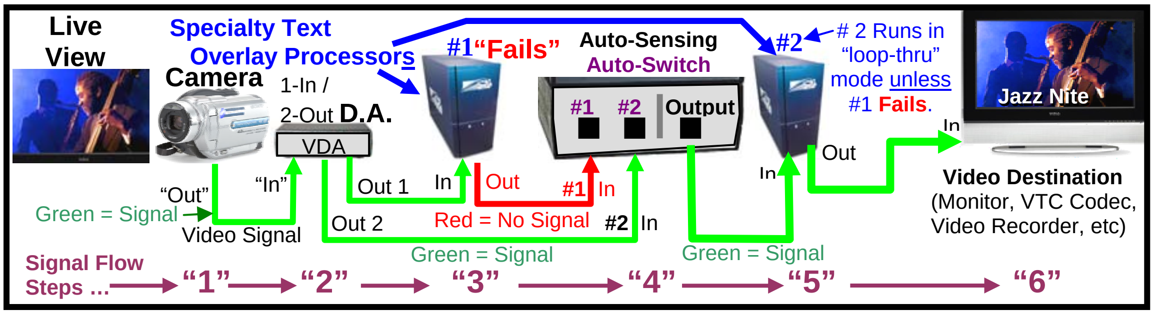
If *(as in this example)* the "Video Destination" unit (5) (Monitor or the VTC Codec) was receiving and further processing / displaying / "transmitting" the same graphic or visual information no matter whether the Specialty Text Overlay Processor was "in-the-signal-flow" or not, *then* the resulting video signal / image sent to the monitor or VTC codec would only experience a [possible] very short rapid "blink" during the switching-interval. **Most end-users would not detect any problem, until such time as an end-user or technician might attempt to invoke some advanced capability that only the (now off-line / failed) Specialty Text Overlay Processor could provide - - - In this example,** in the event that someone actually attempts to invoke a text-label to appear on top of / as an overlay-to the video images, this **"text overlay" cannot happen because**, as shown graphically & textually above, the unit that performs this function has "failed" and has, therefore, been auto-bypassed by the video **auto-sensing auto-switch**.

In other words – when *every* unit is working properly, the signal goes from **Step "1"** to **"2"** to **"3"** to **"4"** to **"5"** uninterrupted. **If something like the Specialty Text Overlay Processor ("3") were to "Fail", then** the signal goes from **Step "1"** to **"2"** to **"4"** to **"5", with step "3" automatically bypassed as a result of** an **automatic input-output change within the Auto-Switch (step "4").**

As noted in the discussion paragraphs above, and in previous documents related to “Redundancy”, the *perception* of the typical end-users will likely be that the complete system is operating, but with some technical “limitation of capabilities*”, though it is entirely possible that many end-users will overlook any minimal missing element entirely, believing that there is actually no “failure” anywhere within the system(s) *(even though we know here this is not the “reality”)*. *The “limitation of capabilities” refers-to functions that were provided [and are no longer provided] by the now “**Failed**” unit(s). In this case the “unit-capability-limitation” is the bypassed **Specialty Text Overlay Processor**.

In a different scenario, if the **System Designer(s) have determined that, in our example here, the Specialty Text Overlay Processor is absolutely essential** to maintaining the communications at a level that is satisfactory to the user community, **then** the auto-switch might not act to provide “bypass” of both a failed unit and the function it provides, but, rather, to “bypass” **only** the failed physical unit in favor of sending the signal through a hot-standby component of similar or same functionality.

(**Please Remember** – This is “**an**” example, **not** “THE ONE AND ONLY UNIVERSAL” example for all situations.)



In this example [above], the “Primary #1 Specialty Text Overlay Processor” (3) *[for the purpose of this explanation, let’s once again state that this processor provides the ability to display a momentary electronic text label along the lower border / edge of the video for the purpose of providing location or other subject information to the viewer]*, as before, **suddenly goes “off-line / fails”.**

If we had not planned a solution for this problem*, then** we would lose the Camera image **(1)** as fed from the Camera - to the Video DA **(2)** - to the Specialty Processor **(3)** and then out - to the Auto-Switch **(4)** – to the Video Destination **(6), since the video would “stop” at failed-step-“3”** and never make it to steps “**4**”, and “**6**” (and, of course, we assume***, for the purpose of this paragraph in our discussion, that, since no plan was made for “failure” in the **Primary #1 Specialty Text Overlay Processor**, the **Secondary #2 Specialty Text Overlay Processor** (#5) would not be installed and part of the flow).

***HOWEVER** - As a result of our design setup* **where we have planned for redundant failover of certain system elements / components, and within which we have [this time] now determined that the use of a Specialty Text Overlay Processor is essential to continue the communication** [for whatever reason – we are **not** concerned, during this discussion, of “**why**” **this is the case** within the confines of an undefined user application set, and we are not entertaining any opinion here as to whether text overlay is critical to all visual video communication / videoconferencing or is not],

in this case the Auto-Switch auto-senses the loss of this Primary (#1) Input signal

from the Primary #1 Specialty Text Overlay Processor and instantly switches to the Secondary (#2) Input (the Camera image as fed from the Camera **(1)** direct - to the Video DA **(2)** and directly out - to the Auto-Switch **(4)** – to the **Secondary** (#2) **Specialty Text Overlay Processor** **(5),** and [finally] - out to the video Destination (**6**)).

PLEASE ALSO NOTE: IN ORDER TO "MAKE THIS WORK":

Along with this "**Switch**" that auto-bypasses the "failed" **Primary #1 Specialty Text Overlay Processor**, **we [likely] need to send a special command from the Remote Control System** ***(this is an example of the use of "Background Control System Intelligence" applied to something other than presentation of an End-User Interface)*** **to** the **Secondary** (#2) **Specialty Text Overlay Processor**, commanding it [the **Secondary** (#2) **Specialty Text Overlay Processor**] to **change** ***from*** **"Loop-thru Mode"** [where it had previously been set in order to avoid having text overlay of text overlay, which would result if both Text Overlay Processors were actively functioning to apply text at the same time on the same video signal] ***to*** **"Text Overlay Mode"**. (**NOTE: "How" such a command can be set to automatically trigger** is **not** the focus of our discussion here. Source video image signal sensing, power-load sensing, video synchronization changes, or other elements can be monitored and set to trigger command sets as a result of any changes. Suffice it to say that such an architecture can be [and often is] built into integrated system designs. In this example, **perhaps** the auto-switch alerts the Main Control system that a command is necessary. We leave it to the integration design and programming specialists for the Main Control System(s) hardware and software to determine which mechanisms are ideal when these actions are necessary).

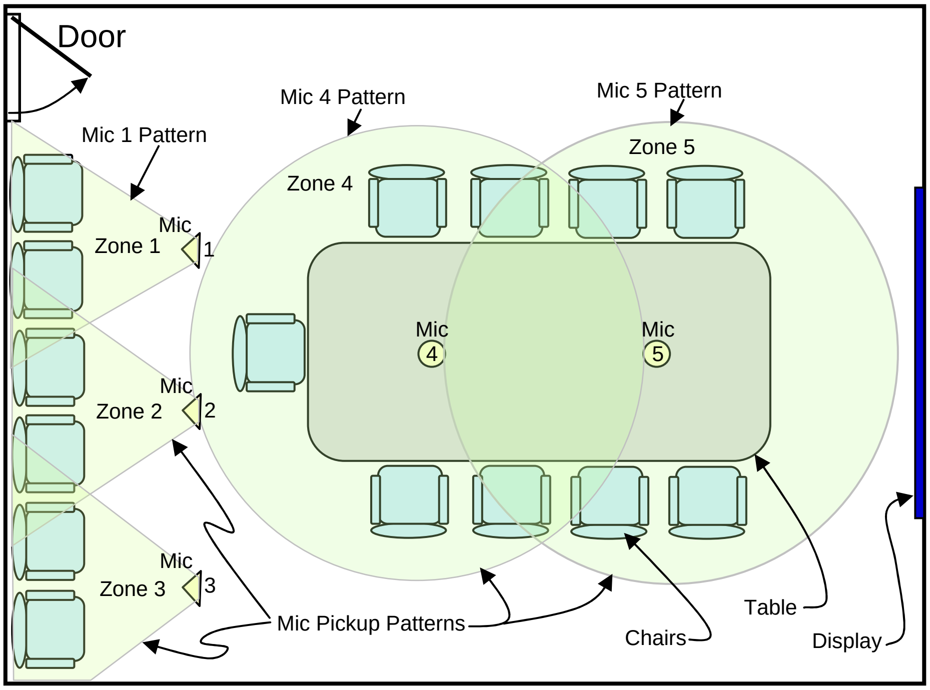
If *(as in this example)* **the "Video Destination" unit (6) (Monitor or the VTC Codec) was receiving and further processing / displaying / "transmitting" the same graphic or visual information no matter whether** the **Primary #1 Specialty Text Overlay Processor** was active "in-the-signal-flow" or the **Secondary** (#2) **Specialty Text Overlay Processor** was now invoked and active "in the signal flow", **then** the resulting video signal / image sent to the monitor or VTC codec *might* only experience a [possible] very short rapid "blink" during the switching-interval. **Most end users would not detect any problem or any "switch" from a failed unit to a secondary unit of identical capability. In this example,** in the event that someone actually attempts to invoke a text-label to appear on top of / as an overlay-to the video images, this **"text overlay" will still happen** [it will still be possible to add / change / delete / or modify the text that appears on top of the video image] **because,** as shown graphically & textually above, the unit that performs this function and that has unexpectedly "failed" has been auto-bypassed by the video **auto-sensing auto-switch** and ***an identically capable unit has been switched into the signal flow***. **In other words – when *every* unit is working properly**, the signal goes from **Step "1"** to **"2"** to **"3"** to **"4"** to **"5"** [with #5 running in loop-thru mode] to **"6"** uninterrupted. **If something like the Primary #1 Specialty Text Overlay Processor ("3") were to "Fail",** then the signal goes from **Step "1"** to **"2"** to **"4"** to **"5"** [#5 is commanded to run in text-overlay-mode] to **"6", with step "3" automatically bypassed as a result of** an automatic input-output **change within the Auto-Switch (step "4").**

So far our examples have been centered on video signals and video signal processing devices. Audio is, however, even more important, since it is possible for humans to communicate at a distance via audio without any video, but in VTC it isn't *generally* possible to have it the other-way-around (video with no voice-audio included**).**

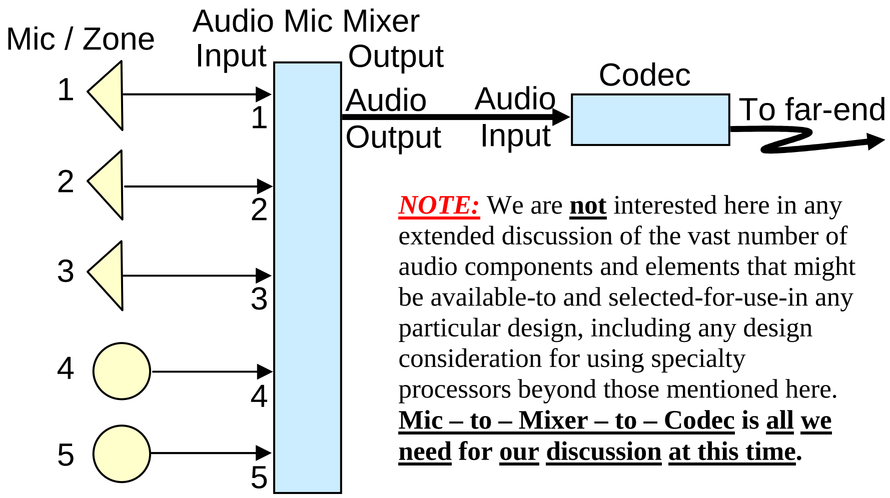
[Please NOTE: Videoconference meetings conducted using only Sign-Language are not considered in the context of this discussion].

Our next example will consider one subtle yet critically important use of Background Control Intelligence as applied in the face of extremely rare yet potentially critical audio component failure. **In order to properly define the application of failover recovery / redundancy** for an audio anomaly (as opposed to failover with full-component complete bypass), **let's first establish a simple predicate design**.

In any videoconference space there is **a** microphone **or** there is an extended array of **some number** of microphones. ***For the purpose of this example***, we will assume that we are looking at an integrated room where there are a number of microphones, specifically selected and specifically placed in order to create an even 'coverage-pattern' throughout the space, enabling people (participants in the conference who are located in this local room) to be clearly and intelligibly heard from where they are typically seated within the space. [**NOTE**: We are **NOT** concerned, for the purpose of this discussion, with the make or model of any microphone, **nor** are we concerned with expanding the discussion to include a detailed explanation of proper and effective audio design for videoconference. That is **NOT** the purpose of this whitepaper. **We are strictly concerned with** the simple goal of complete and even coverage for the purpose of intelligible pickup at a gain-level appropriate to VTC activity. ***The examples provided here are meant as only that* – EXAMPLES** - FOR FURTHERING **THIS DISCUSSION**, **NOT ALL** AUDIO-RELATED DISCUSSIONS FOR **EVERY** ASPECT OF AUDIO DESIGN OR **EVERY** CONFIGURATION FOR **EVERY** SYSTEM FOR **EVERY** DESIGN **EVERYWHERE** AT **ALL TIMES**].



For our purposes, this represents not only a simple microphone layout and the related microphone pickup-patterns (areas of pickup of room-voices by a particular microphone), it also represents multiple "**zones**". Each microphone is individually cabled and patched into a mixer that permits variable independent adjustment to the signals that come into the mixer from each microphone / “zone” **prior-to** combining the signals into a single (1) audio signal that can be sent [output] to the codec for transmission to another location.



NOTE: We are **not** interested here in any extended discussion of the vast number of audio components and elements that might be available-to and selected-for-use-in any particular design, including any design consideration for using specialty processors beyond those mentioned here. **Mic – to – Mixer – to – Codec is all we need for our discussion at this time.**

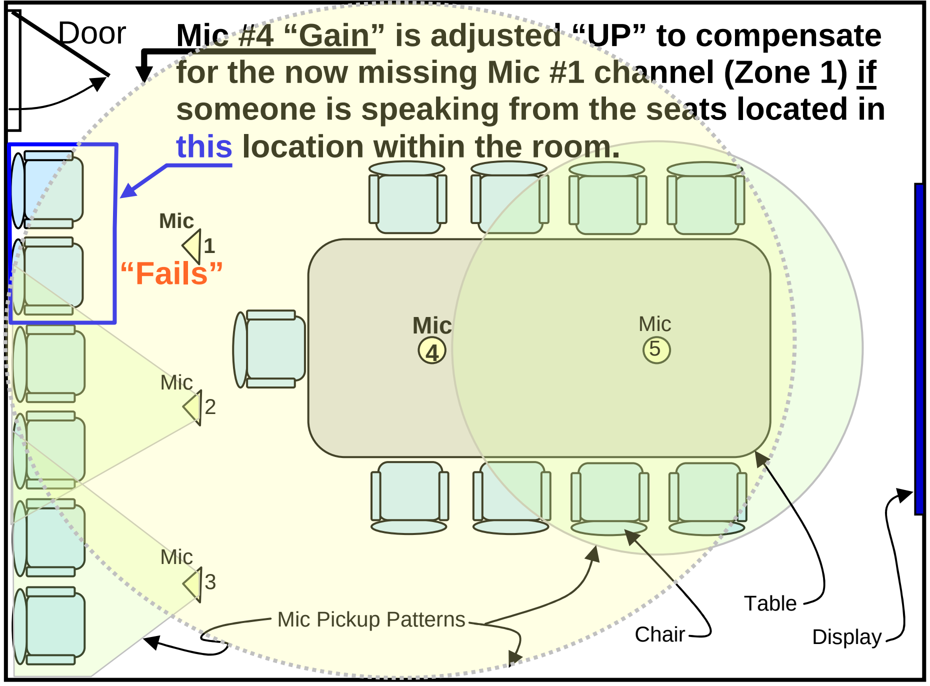
Yes - The essential capabilities expressed by this type of simple layout are quite common, and this allows Integration and Operations teams to adjust the audio as necessary, providing balanced & discrete gain levels with even coverage in a given space.

Additionally, this allows the Operations group to adjust the levels according to typical dynamic events that unfold during a meeting. ***For Example:*** If two room-participants (people who are included in the room for the meeting) in Zones 1 and 2 (this means Mic 1 and Mic 2 respectively) engage in a random and non-primary side-bar-conversation, the Operations personnel *could / might* reduce the pickup volume for those microphones [or mute them completely] in order to prevent the side-conversation from interrupting the main audio exchange, until such time as the participants in Zones 1 and 2 were no longer engaged in disruptive side-bar but were prepared to engage in generally productive conversation related to the meeting and to the benefit of every other participant.

Our discussion here, in this paper is, however, related to “failover” and “redundancy”. Let’s expand our discussion, using the same simple room layout, through an adjusted example. Suppose, for instance, that one of the microphones (Mic #1, picking-up anyone seated in the two seats within the pickup area or “Zone” for that mic) suddenly “failed”. [Or, similarly, suppose that the input port for the Audio Mixer for Mic #1 suddenly “failed”.] **In that case we could anticipate that, suddenly,** anyone seated in the two seats located in the Mic #1 Pickup “Zone” will then not be clearly heard. (Leaving aside a lot of technical audio-discussion, **let’s simply accept** that their voices will, *most likely*, seem to ‘listeners’ at the far-end(s) of the call, to have become very faint, as opposed to disappearing altogether, since their voices might be (and generally are) still minimally picked-up, according to the possibilities of our simple diagram, by other room microphones, in this example - by Mic #2 or by Mic #4). ***With proper planning, however,*** and **assuming that the Mixer was of a type that allowed for discrete analysis of the individual audio signals and**

levels by an external device like a Master Room Control System, once the signal from Mic #1 fell below a specified level, another microphone *could* [*might*] have it’s pickup volume or “gain” ‘automatically’ increased above the original “normal” level in order to make up for the deficit (in order to extend the coverage of this microphone to cover the area that was suddenly left uncovered by the [now] failed microphone).

Naturally, the overall “balance” of the microphones might *(probably will)* now be “uneven” (possibly out-of-balance to an extreme, since, if we advance or increase the pickup gain on one microphone in order to adequately hear an atypical ‘other Zone’, then the people in the assigned Zone for that now increased microphone will appear to be much “louder”), **but this effect, too, can be ameliorated** through adjustment of the upper threshold values for the newly adjusted microphone gain and additional adjustment to [possibly] all of the microphones within the space (more on this statement below). A “limiting” circuit can be invoked, and, if it is a circuit that has a very fast “attack-and-release” when handling the gain & upper volume level, then the mis-match that is created within the pickup area of Microphone #4 below can be minimized, with the level ramped “open” or “up” when someone in Zone 1 / failed-Mic #1 is speaking, and “down” or “attenuated” when someone in original Zone #4 / Mic #4 is speaking. ***For instance:***



(**Note that** the capability and discrete logic control for the Mixer must allow for this, and that will mean an advanced type of Mixer component will be required). **ALSO NOTE - We need to introduce (actually – “expand-upon”) a side-comment of deeper “audio” necessities, in an effort to illustrate that this type of “simple” activity is actually more complex and convoluted than it may first appear:** In this scenario, not only would a “limiting” circuit need to be invoked & commanded to manage Mic #4, but Mic #2 and Mic #5 may [probably will] need to be adjusted “down” (attenuated) or [possibly] even “muted” entirely as part of the attempt to “even-out the gain structure” as a result of the failure in Mic #1 and the use of Mic #4 to make up or “cover” for the failure-loss.

ADDITIONALLY – THIS EXAMPLE IS SIMILAR TO THE PREVIOUS EXAMPLES OF "REDUCED FAVORABLE PROCESSING" OR "LIMITATION OF CAPABILITIES": **It is critically important to note that *even though it is possible*** to construct and integrate a microphone-and-mixing audio system to perform this type of dynamic "recovery and adjustment", **and *even though this can be designed*** in such a manner as to ensure that everyone is clearly heard at all times, even during a few milliseconds of system-recovery, the **audio quality will most-likely suffer**. It's not that there will not be enough "gain" or volume" – there can and [generally] will-be. It is not a question of maintaining "volume". The difficulty will [often] be in the overall **perceived *quality* of the sound** in terms of "fullness" or "richness", since, as any audio specialist knows, the further away from the pickup device we get, the less "rich" and more "hollow" our voice becomes, as a result of the nature of artificial pickup using electronic microphone systems.

By increasing the mic "sensitivity" (***o.k***. - we are not *really* increasing the mic sensitivity – *this is set or established by the type/make/model of the mic* - but we have ***indirectly appeared to have done this*** by dramatically adjusting the input level from a particular microphone "UP" on the Mixer to a dramatically higher level) **this action is generally going to negatively impact the Signal-to-Noise (S/N) ratio for the system and room, and [most likely] by a considerable distance.** The audio entering the [now extremely "hot"] microphone [Mic #4] will appear to have [will be audibly perceived to contain] more ambient "noise" present from the room, and any random user / participant "noise" (paper shuffling, pen tapping, etc), or any fan-noise or other equipment noise present that has been generated by the HVAC system (**H**eating **V**entilation and **A**ir **C**onditioning) or other facility services equipment, or any noise that might be generated by the room AV (fan-noise from a projector or plasma display, for instance), will now appear to be more "present", and will compete with the spoken word for our attention as the "primary" audio we hear from this space. **As the separation of Signal (participant voices) and Noise (background noise) is lessened** (the ratio gets closer to 50-50 or worse), **intelligibility will suffer dramatically**. We will, however, be able to continue the meeting (especially if the participants act responsibly to control any extraneous and unwanted noise), hearing the participants at a **minimally** acceptable level and quality [at least], until such time as the meeting can be completed and the "failed" audio element (the mic or the port on the mixer) can be attended-to and repaired, and the system returned to the originally designed operational parameters (gain settings and pickup coverage area(s)). **This**, in point of fact, **is *generally*** [not "*ALWAYS*"] **preferable** to the alternative – the complete loss of the vocal contribution from any one or more of the participants in the meeting room. **HOWEVER** - ***The result might be*** a decision during the design phase to duplicate the microphones as the mechanism of "redundancy" rather than rely solely on restructuring the gain settings in the mixer, in an effort to maintain higher "quality" in the audio, separate and apart from maintaining adequate volume or gain. This would be the essential approach **if it is determined that** audio is not only required, but **ideal intelligibility** is also required at all times by this group of end-users, and any loss of intelligibility or apparent increase in "noise" levels that accompany the voices (any degradation to the S/N ratio) is not permitted (for whatever reason).

Now – Having stated that **there is a critical decision layer for the *System Designer*** related to the determination of when, where, for which elements and in what ways there might be some use of automatic bypass and rerouting of signals, based on the use of auto-sensing devices and/or commands from a Central System Control Frame, it is important to see if there is any "**general** list" *[that's "**general** list", **not** "carved-in-stone-always" list]* that can be compiled that *might* be applied to *most* integrated systems. ***This is a delicate and difficult process for which this writer has no "magic perfect 100% applicable list"***, since each and every integrated system is unique, and **any attempt to make a "Master List" must** be made with the full and complete knowledge that any **application of such a list does NOT dismiss the Designers and Control Specialists from their responsibilities** for careful analysis and assessment for exceptions to any such list within each system they design.

When considering the elements that ***might*** be candidates for 'bypass and failover to hot-standby units of same or similar capability', we could be well-served to **consider that videoconference** real-time visual communications, at the essential core of the activity, **requires *seeing and being-seen***, ***hearing and being-heard***, and ***stable sustained connections*** between endpoints. [**In addition,** it is essential for the elements of **source power** and **system control** to be included on this list. **If there is no power**, then the electronics will not operate, no matter how much redundancy or failover architecture is applied. **If the system cannot be simply, easily and quickly controlled,** then the components cannot be made to work in unison in any dynamic manner to achieve and sustain a connection or provide an exchange of audio and video information that is of an adequate and useable format & level for the human beings engaged in the activity.]

Within the categories listed here as *potential* candidates for bypass and failover to hot-standby, it is also necessary to consider the depth of negative impact that ***might*** be experienced by any particular failure. **While this author takes no position or posture on any system aspect or component capability for the purpose of this specific discussion, *common-sense leads us to believe that there are some obvious candidates* for 'redundant failure to hot-standby' and other less obvious candidates for this same level of critical treatment.** Analysis of this sort will [often] guide us in the **reasonable** selection of those elements and functional aspects we select as "elements to be made fully redundant and automatically recoverable" in any system(s). In the earlier examples above, for instance, the loss of "Text Overlay" at the bottom of a video image may, in the context of many commonly held meetings, not rise to the level of **"***critical and necessary, without which the videoconference communication will completely come-apart***"**. In our audio example, however, the loss of the ability to actually hear or be-heard will almost universally create an intolerable and completely unusable situation for the participants. **As previously noted** – each and every system is and will-be unique, as will each and every application for which the videoconference and visual communication technology is integrated and utilized. Only a thorough and careful analysis of the applications, constructive discussion with the end-user community, and reasonable common-sense creation of some "bounds of limitation" within a hierarchy of required elements can generate a final "failover to hot-standby redundancy" list of candidate technologies and capabilities. This is all **done by the System Designer** or Design Team.

For any systems that are or will-be designed for Videoconference and Visual Communications, the list of bypass & redundancy considerations in '*automatic failover bypass to hot standby*' should include, but **not be limited-to**, the following items:
1. **Primary display devices**, including CRT, LCD, Plasma or **any form or format** of Projection device(s), and any associated specialty display or video signal processors.
2. **Primary camera input** devices and any & all critical data source devices.
3. **Primary microphone audio input** device(s) and any & all processor(s) essential to their operation (mixer, combiner, echo controller, etc) for out-bound sound.
4. **Primary audio output** speaker device(s) and any & all processor(s) essential to their operation for in-bound sound distribution (amplifier, echo controller, etc).
5. **Network connection, including** any & all edge-elements that make connectivity into and across the network possible. Such elements would include network interface or access points (CSU/DSU, IMUX, Switches, Hubs, Routers, or other similar elements), and Network Services (ISDN Carrier Services and / or Ethernet LAN or WAN backbone connection and access routing).
6. **Remote Control Access devices** (hard-button and/or LCD or other touch-screen devices) **and primary or "main" Control Frames** (where the individual device control connections are made and the primary software source operating system is stored, accessed, manipulated and "commanded" from the Control Access Devices).
7. **General Power** (from the facility or electrical energy supply source) and **Specific Power elements** (specialty transformers, distribution elements or other items that might be uniquely essential to one or more individual system components) for all of the elements listed above.
8. **Any other intermediate or terminal processors** that are deemed to be REQUIRED AND NECESSARY, **without which the videoconference or visual communication will be diminished to a level of ineffectiveness that automatically establishes that the activity, in light of any failures in these components, cannot and will-not take place.** These will be determined, generally, by an analysis of the nature of the communications that emanate from a space or within a particular group, the requirements any individuals or applications might impose for specific aspects or elements of signal presence and handling (if all meetings, for instance, require the display of PC-based data, then a redundant PC *may* be required within the space), and required 'quality' levels.
9. The **overall common-sense limitations** that *might* be **imposed by the balance of** "fully redundant systems recovery" **with** "cost-effective systems deployment and use".

PLEASE NOTE: **This concept of "failover to hot standby" and "redundancy" generally requires more capable / higher-performance components.** ***In our audio example above, for instance,*** a simple analog audio mixer that is comprised of nothing more than 5 input ports, one output port and manually manipulated rotary volume adjustment knobs, will not permit the level of control and intelligent polling that will be required in order to maintain the outbound audio and for adjustment of the multiple signal elements in any dynamic fashion via artificial electronic Background Control Intelligence. **This means, of course,** that the components will generally cost more ("**brains**", also known in this discussion as "component intelligence and processing capabilities", will **cost money).** ***Care must be taken*** to qualify individual electronic components not simply on "cost", but also on native and necessary performance (signal handling and amount of "brain-power"), the ability to communicate with each component via external command and control elements in order to more discretely and easily manipulate the settings or adjustments in real-time, the ability of any unit to report "status" to the Master System Control Frame and, finally [and potentially], the ability of individual components to "assist" the Master System Control Frame with commands to other units or to "assist" by issuing internal "state" commands to self-adjust internal configuration(s).

Likewise – the ***cable*** and ***power*** required, along with any added burden on the ***HVAC*** for system cooling [and acoustic isolation to limit any negative impact on the general ambient noise-floor within the space], **must be considered in detail PRIOR-TO purchase and implementation** of the actual hardware components within the system flow. **Too often** the cabling, especially the cabling that must be pre-wired in ceilings or walls or added to certain "drop" and distribution points, is completely overlooked, and original facility conduit capacities and HVAC cooling capacity [with minimally noisy air exchange in the air handling] may not permit cost effective after-the-fact addition of some "failover redundancy" elements. **Remember** – good planning here is based on the fact that the number of power and network 'drops', and the capacities of the power and network source supply and distribution [conduit, wire-ways, etc.], **almost**-always increases over time. Planning and installation of "just enough to 'get-by'" at the time of the original renovation or system installation / upgrade will **nearly** guarantee that, within a year or two, additional cable and power drops will be needed, and if there is not adequately planned initial deployment of conduit and inner-duct in the original designs, along with accommodation and placement of spare pull-strings and access traps, then [often] costly additions will have to be made in the future, frequently in such a manner as to take the room or systems 'out-of-service' for some extended period of time.

Likewise: ***Contrary to input that this author has recently received*** from one high-end successful Systems Integrator, and two enthusiastic Designers (at two (2) different locations), "Failover to hot standby" and "Redundancy" generally requires more equipment, and even more advanced equipment *(even beyond the cabling and other facility elements we listed in the paragraph above)*, not less. Flexibility and survivability come at a price, in $$$ *and* complexity.

Control Layer – to – Control Layer Communications and Real-time Automatic Configuration Adjustment of the possible breadth and depth of application-specific End-User Options.

This is, quite possibly, one of the most difficult areas for this author to attempt to define [here in this paper], comprehend [on the part of this author and all persons involved in the process of defining, developing, designing and deploying communications technology such as that used for videoconferencing], and then plan-for, design-for, integrate and install. What we are **about** to discuss is a set of highly complex processes that [**frequently**] have far-reaching consequences in the design and application layers of the development and deployment model of the Master Control Systems used for visual communications and videoconference.

To begin this portion of our discussion, let's make note of **a few** [that's 'few'…not 'all'] **fundamental historically common elements in Control System Design,** several of which we have touched-on already in Part 1 and the initial portion of Part 2.

1. ***Most*** **Control System Design takes as a focus** the need to gain control over the fundamental operating functions of individual electronic elements, for the purpose of obtaining optimum and complete performance of those elements. We "control" the video projector for the purpose of making ideal adjustments to that projector and accessing subsystems of that projector [we might simply need to control the source-input switching capability of the multi-input interface card, for instance]. We "control" units such as the videoconference codec for the purpose of telling it to take certain actions, such as dial another codec, change the negotiated resolution of the video or manipulate the echo-handling circuitry in order to reduce or eliminate audible echo.
2. ***Most*** **Control System Design takes as a focus** a need to provide to the various users of the systems idealized interface to those systems. We attempt to "customize" the experience based on what we know of the end-user, our assessment of what the end-user may require in order to functionally and efficiently use the technology, and the general need to operate or make adjustments to the technology. Within a videoconference codec, for instance, a non-technical end-user may only need access to a phone directory, since all they might need to do is dial to another location. A technical end-user, working in the technical support or maintenance area, using the same codec might need access to the dialing functions and deeper access to items such as diagnostic tools and internal configurations resident in the codec. A technical end-user may also need to restrict adjustments to the configuration of the codec and, since certain mis-adjustments could create a negative outcome, we might look to use the Master Remote Control Interface as a check-point by which only qualified technicians can make any adjustments, while non-technical end-users are denied access to selected adjustment mechanisms.
3. ***Most*** **Control System Design takes as a focus** a need to control local elements within the physical space where the integrated system resides, even though we have indicated, in the document above and in a previous document related to Redundancy, that the "intelligence" of the Control Systems can be used for activities and purposes not directly related to Control System access and input activities of the end user, and even that those applications can be for management of the local or remotely integrated system components. *In short –* ***Control has traditionally been a "localized in-room activity".***

In this current portion of our discussion we are going to take a wider yet more intricate view of the possible uses of the Control Systems Intelligence. We have now arrived at a place where we will state that **Control Systems used for visual communications / videoconferencing** at a distance **can make for a much more powerful solution if they begin to communicate** not simply with the component systems and sub-systems within their own localized integrated domain but if they also communicate **with one another**, operating within a global set of guidelines and rule-sets to reach idealized "states". **We further state that** the control systems **could** actually even ***begin to*** manipulate their own [and each others] user-interface options based on conditions and states of distant systems, at times offering [possibly] fewer options and at times offering broader options to their own localized user community based on these remote-originated communications.

There are several elements that must be addressed for this approach to be both effective and beneficial to the ultimate ***optimized flow*** of communications **between the human beings** engaged in person-to-person videoconference based communication, **three of which rise to the top of the list** in this discussion paper.

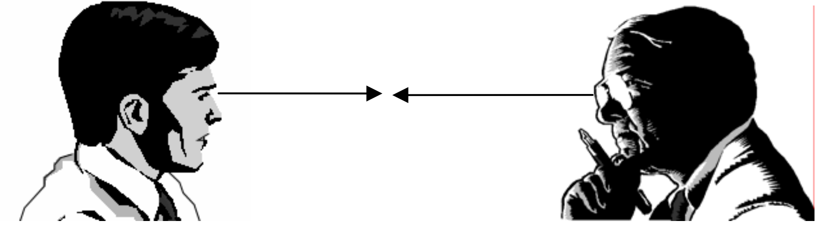
1. A set of "global requirements" for optimized visual and auditory states must be defined, agreed upon and established.

2. A set of real-time adaptive hierarchies must be defined, developed and agreed upon that will account for the organizational hierarchy of information and decision flow.

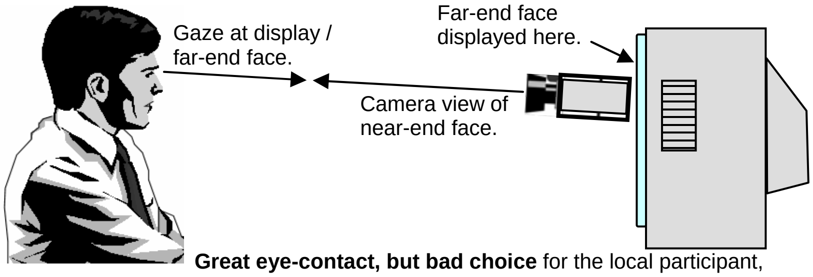
3. An agreement must be reached as to the methods or mechanisms for Control System-to-Control System communication, and this must take into account point-to-point communications (two terminal devices) and multi-point communications (more than two terminal devices).

We begin with #1 [on this page above]**: A set of "global requirements" for optimized visual and auditory states must be defined, agreed upon and established.** We must define, ***ahead of time***, the **exact elements** that are **considered "essential" to idealized natural human communications** as related to ***specific*** end-user applications of the communications technology and ***specific*** system design integrations.

For instance**:** ***The Parallax Problem in Videoconference.*** [Though **not** the focus of this discussion *(this paper is* ***not*** *about maintaining eye contact in videoconference),* we are providing here a short explanation of this issue for the purpose of establishing how we might use the Control Systems Background Intelligence to help this aspect of videoconference communications.] It is a well-studied fact that "face-to-face" contact does not result in the most powerful communication via videoconference, "**eye-to-eye**" contact does. **It has long been well-documented that eye-contact is essential to effective face-to-face communications in any form, including videoconference-based distant video communications.**

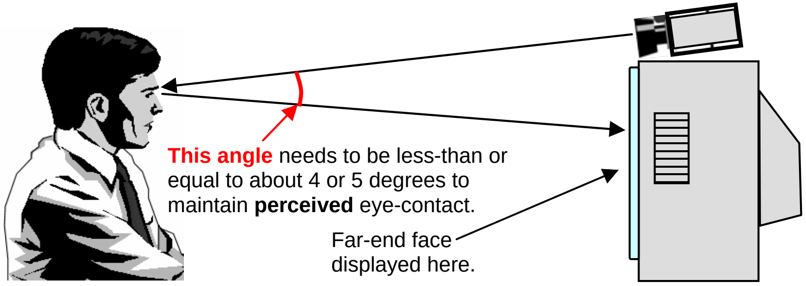


In the area of "levels of interaction", for example, we know that **as eye-contact increases between people, the amount and quality of the interactions will go "up"**, often times dramatically so, **and this is especially true during a videoconference**. Eye-contact has long been *one* element of assessing the veracity of interpersonal communications. Everyone is familiar with the phrase "Look me in the eye and tell me the truth", based on the theory that it is more difficult to lie to someone if you are looking them in the eye, and easier to detect if someone is lying if they have difficulty maintaining eye-contact. The extensive work that has been [and is being] done on improved "eye-contact" via video, using various mechanisms and methods, indicates how important this is to effective distant communications [videoconference communications]. ***As a result of knowing how important this is we can also begin to make some assessment or judgment of the space and system elements that are intricately involved in either supporting or degrading the quality and/or level or amount of eye-contact.*** "Parallax", for instance, tells us that we need to locate the camera* *(* this represents the 'eye-location' in our own room of the people who are at the far-end of the call)* as close to the center of the gaze of the local participant as possible, if we are to maintain eye-contact through artificial electronic means with the person or persons at the remote site(s). Obviously, people (or "human animals") will generally look at the video image of the "remote" face(s) of others arriving into their own room as displayed on their local video display. Based on this, **the *theoretical* ideal placement for the local camera** that will send our local image of people / faces out of the room **would be at dead-center of the local display** [the unit that is locally showing the "remote" people or face(s) of others arriving into the local room].



Great eye-contact, but bad choice for the local participant, **since the camera "blocks" some of the video display**…

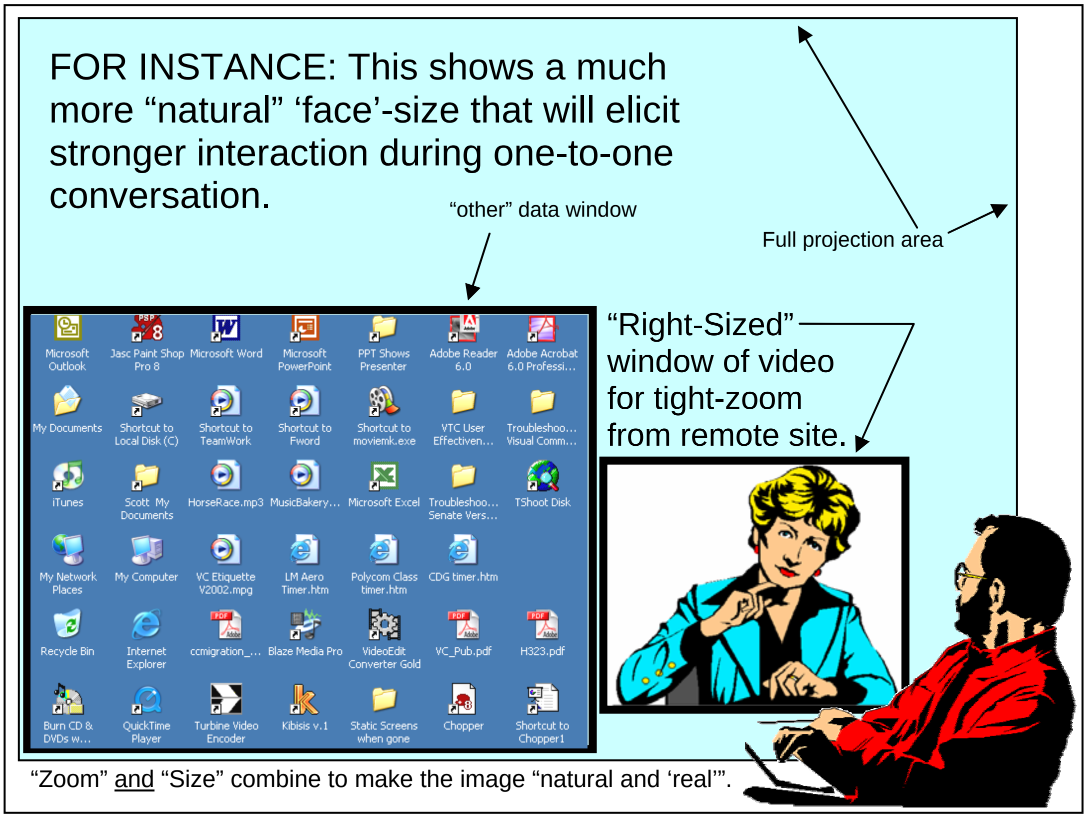
This, as we see in the graphic above, is not entirely practical, since placing a camera in front of the video display means that local viewers of the display can no longer see the inbound images unhindered. Common practice is often, then, to place the camera as close to top-center of our own display as possible (seen below). **Scientific studies have shown that if we keep the angle of the line of our gaze towards the local display and the line of the camera view of us less-than or equal to about four-to-five degrees (horizontal *and* vertical)**, then the people at the far-end of the call will ***feel*** like we are 'looking them in the eye', even though we are not [***we are not*** *looking at the lens of the camera – 'the "eyes" of the far-end participant' –* ***we are looking at*** *their video image as sent to us by their camera and displayed on our local video display*].



O.K. **–** Having established that eye-contact is critically important, and how to best maintain this, it is essential that we consider other fundamental elements that **idealize** this in an electronic format. In order to keep our discussion brief, ***we will look at just one*** (1) of these elements – **image size**. **Simply stated**, the further away from the camera that we get, the more shallow the Parallax angle becomes. This is a good thing. We already stated that we want ***less-than*** *or* ***equal-to*** four or five degrees. However – the further away we get from the camera, the less **perceived** “detail” we have in the actual image being displayed. To compensate for this, we generally “zoom” the camera into a tighter view. The level of “zoom”, however, is not strictly determined by the need for image detail of the eyes. We stated, in the beginning of this section (#1 – “*A set of “global requirements” for optimized visual and auditory states must be defined, agreed upon and established*”) that we need the communications to be ‘**optimized and idealized’** for “**natural**” human communication. **Eye-contact begins to help us to *feel*** that the communication is “natural” / “real”, but “**Size**” **is also a critical element**. We are most comfortable, as human beings, when the size of the faces / heads of the person or persons at the far-end, as displayed in our own room, are approximately “**right-sized**” for our distance from their display on-screen. In videoconference communication, as eye-contact goes “up” ***and*** as “right-size” or “natural feeling” related to size goes “up”, so does our comfort level and, hence, so will our level of interaction. **FOR EXAMPLE: If we are sitting 8-feet away from a face that fills a 10-foot tall screen, we don’t feel like we are “talking with someone who is sitting across the table”. We feel overwhelmed and, consequently, limit our interaction.**

We see the “eyes” clearly, but this does not promote “one-to-one” conversation…

In order to accommodate this potential conflict, we *might* use their camera to “zoom” close enough for good detail of their eyes, while also using an image processor at our location to display that remote-person not on our entire large-screen display, but in a size that is more “realistic” and “natural” or comfortable for the local viewer.



FOR INSTANCE: This shows a much more “natural” ‘face’-size that will elicit stronger interaction during one-to-one conversation.

“Zoom” and “Size” combine to make the image “natural and ‘real’”.

***Getting back to our topic, we are reasonably forced to ask:* Just what does “Control System Background Intelligence” have to do with this manipulation and “setting” of camera-zoom and image-display-sizing?**

Simple. We have two basic options for establishing the proper display of the information.

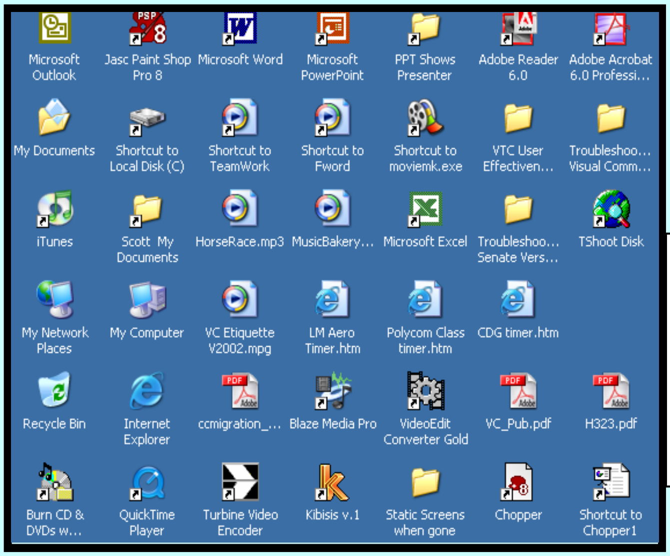
1. **Within our own local room**, and using our own Control System, we can manually set-up the images for “right-size” display. We will need to ask that the far-end set a particular “zoom” on the camera, and then we will need them to agree not to change that zoom level, or we will have to pre-set infinite “right-sized” layouts. We can then attempt to quickly & **reactively manually select different display sizes for their inbound video** at our end whenever the far-end changes the zoom setting of their camera.

OR [using “Background Intelligence”…]

2. We can establish the “right-size” variable range [locally] based on the available range of camera zoom increments [at the remote end], then establish a communication link between our own Control System and the Control System at the far-end. Anytime the “zoom” changes at the far-end (based on changes to preset selections or manual adjustment to the zoom), their Control System can communicate the specific details of this change to our Control System, and our Control unit can then command our local video image processor to make the “re-sizing” adjustment on our own display in near-real-time for us, without our input. This means that we can use **Background Control System Intelligence to replace human manual reactive commands.**

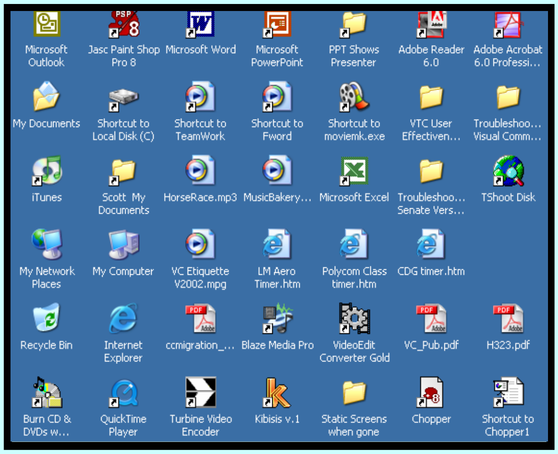
***FOR INSTANCE*:**

THEIR zoom setting is too far away to be of use to us if we are going to look at them and their PC data on OUR screen. We have no image detail or sense of eye contact or “right-size” for natural human communication with their site participants.



This [above] has an “incorrect” remote zoom for ‘natural’ eye-contact interaction, so the local controller would send an instruction to the far-end controller to instruct it to change the remote camera zoom, possibly to something more like:

In this view we have commanded a change in THEIR zoom level and also changed OUR image ‘window’ of THEIR video [and the size of the PC data display] to create an optimized ‘right-size’ in order to enhance the feeling of natural human communication and interaction.



Likewise, the ability to have Controller – to – Controller intelligent optimized "adjustment" *can* be synchronous in terms of state-information exchange and 'commands'. ***Suppose*** we were to select a different display size at our end. It is possible that we could then have **<u>our</u> Control System communicate** this change **to the far-end Control System**, and **<u>their</u>** system could automatically command the camera PTZ controller to "zoom-out" or "zoom-in" in order to maintain idealized "right-sizing". **This is but *<u>one</u>* example of Controller – to – Controller intelligent optimized "adjustment".** There are many, many more. These ***might*** [*or* might-<u>not</u>] include, but aren't limited-to, the following types of examples (*<u>Note</u>*: an extremely **<u>limited</u>** set is offered here purely **for our discussion**, and <u>infinite</u> <u>other</u> <u>scenarios</u> <u>are</u> ***possible*** for any given design):

Control System – to – Control System Background Intelligent adjustment to:

1. *the overall "send-gain" of the far-end audio* based on the number of occupants in our own room, given the assumption that more people create higher noise levels and this means increasing the gain in order to maintain S/N ratio at optimal levels.

2. *the image size of our video display of inbound information*, based on the preset zoom selected at any one time, with a lock-out feature established based on the number of sites that might be engaged in the call. If multiple sites, perhaps no location would be permitted an extremely wide or distant shot, in an effort to maintain optimized natural views of all locations on a single screen (*simultaneous presence*) at our end of the call.

3. *the image size of collateral information*, based on the anticipated detail of the information sources being used at the far-end (larger images for PC data and smaller images for document camera imaged materials, with smallest reserved for DVD playback from the far-end into our room).

4. *increase the brightness of the camera images they are sending to us* if our display device suddenly begins to lose the ability to run at full-brightness for display of their images (perhaps we have a dual-lamp projection system that relies on both lamps operating at all times and one of the lamps fails, requiring us to continue at half-brightness for some period of time).

5. *the choices for cameras or camera presets based on our own display configuration at any one time*. For instance, perhaps when the far-end decides to use a document camera, the main camera is re-set to a PTZ (Pan-Tilt-Zoom) on the person using the document camera device, and a command is sent to our local system to engage a second video stream, showing both the person who is using the document camera in one 'window' or on one screen and the document camera image in or on a second, and both images are right-sized for the [now] more limited available total space on our local screen, with additional control given to us [that is not normally granted] in case we need to re-size the document camera information.

Note: The above examples, <u>absent</u> specific applications and configurations, may or may-not apply to any systems that you are presently designing and programming.

As previously stated, only a thorough exploration on the part of the Systems Designer and Control Systems Designer and Programmers of the **infinite** iterative option **possibilities** will yield a complete set of **Control System – to – Control System Background Intelligent adjustment** of any elements within a system or set of systems.

We may now move ahead with #2 from page #15 above: A set of real-time adaptive hierarchies must be defined, developed and agreed upon that will account for the organizational hierarchy of information and decision flow. We mean by this that we must define, **ahead of time**, elements like those we have just discussed as "options" for design and programming into the Control System Background Intelligent layer, based on **a hierarchy of needs and a determination of "command" and "obedience"** within any multi-Controller configuration.

All issues related to this *cannot* be fully explored within this short paper, but we can define ***some*** predicate questions to be asked prior-to entering into most of the design and programming decision layer related to this type of advanced control mechanism.

To help with this it *might* be useful for the Designers to explore the following:

1. **According to the applications for which videoconference is being utilized, what are the "make or break" elements of optimized video communication** [video signals, audio signals, source selection, camera control, etc.] that are dependent on configuration states at both ends of the connection?
2. **According to the desires of the end-users, and their perception of "optimized and idealized video communications", what are the "make or break" elements** of optimized video communication [video signals, audio signals, source selection, camera control, etc.] that are dependent on configuration states at both ends of the connection?
3. **How discrete and granular can the inter-communication be** between the different Control Systems at each of the endpoints in a video communication **as determined by the capabilities of the individual components and the flexibility of the integration** found at each location?
4. **How discrete and granular can the inter-communication be** between the different Control Systems at each of the endpoints in a video communication **as determined by the amount of variable change within a given interval** required to maintain optimized performance without overwhelming distraction to the end-users (as a result of the "state" of the systems and the choices offered being altered from moment to moment)?
5. **How discrete and granular can the inter-communication be** between the different Control Systems at each of the endpoints in a video communication that is **point-to-point as opposed to multi-point**?
6. **How will endpoint systems that do not (or cannot) make their Control Systems available for this inter-control communication** influence the ability of the connected Control Systems to properly and effectively provide optimized 'states' for both the 'intelligence connected' and the 'intelligence non-connected' Control System locations?
7. **At what point or within what circumstances are end-user ad-hoc conflicting commands permitted,** and how do the systems accommodate ad-hoc input that does not fall within the programmed Background Intelligent adjustments?
8. *[Possibly one of the most complex and difficult to resolve]:* **When the Control Systems are placed in Background Intelligent communication with one another, which, if any, becomes the "parent" and which, if any, becomes the "child" controller?** Suffice it to say that the example we detailed above, related to Parallax and Image-Size, requires one or more Control Systems (and the endpoint components that they represent at the control layer) to become subservient to another. Which system is permitted to "take control" over another and when? In what way does this happen, and can it be dynamically adjusted or

reconfigured to allow a parent to become a child and a child to become a parent? **It is an inherent conflict** to have one Control system at one location say "You can only send me a certain 'zoom' level because I am using a specific display size for that image" while the Control System at the far-end is simultaneously saying "You need to increase your image size because I am going to send you a very wide camera zoom for this video image / signal". ***The question is simply this*** – **Which Control System takes "precedence"**, under what conditions and for which elements, and how might this precedence be over-ridden or terminated during the course of the active communications, and to what effect?

Finally - Addressing #3 from page #15 above: An agreement must be reached as to the methods or mechanisms for Control System-to-Control System connectivity and communication, and this must take into account point-to-point (two terminal devices) and multi-point (more than two terminal devices) communication applications.

Here we enter into a set of highly complex technical issues that must be addressed and overcome. While **we cannot** [**and will not**] **strictly define each and every *possible* hurdle for all systems** that will employ this control technique, we can define ***a few*** for which further analysis and consideration will be necessary on the part of the Systems and Control Systems Designers and Programmers. **First and foremost,** in order to make use of Control System – to – Control System communication of Background Intelligence, **it is essential for the Control Systems to actually communicate *with one another*.**
(This may seem like an obvious statement. Real-world conversations have shown me, however, that this is a more complex layer beyond traditional system communication than most specialists realize.)

Most Control Systems in-use today have support for various forms of communication with the system [and, in our application, between the systems]. Most have IR and Serial communication ports for the purpose of communicating with the units and components they are programmed to control. Many systems allow Serial communications with the Control System core elements, for the purpose of programming and diagnostic work on the part of the Control Systems Programmers via Hyper-terminal and other command-line structured interfaces. Many also now include Ethernet connections [a few have even shed their RS-232 Serial ports entirely in favor of Ethernet connectivity], not only for communicating with integrated device components but also for permitting communication with the core Control System itself, again - for the use of the Control System Programmer. Within the Ethernet realm these systems support many different methods & protocols of communication - - Telnet, HTTP, HTTPS, FTP, SNMP and other means of data transfer and real-time device & system communication. Most designs **will generally use either Serial** (traditional) communications via [typically] an RS-232 protocol connection, **or will communicate with the system**, and put the system in communication with another remote system, **via the Ethernet port**(s). *Again* – these are the **most commonly available methods** and mechanisms, and most equipment purchased today, and in the near future, will permit the use of either or both of these.

RS-232 Communications with and between Control Systems:

Benefits:

1. Reliable, traditional and still the most common method of communicating with these systems.
2. Simple to physically connect.
3. **Can be used in-band** with a VTC 'call' connection through a codec **by connecting** the RS-232 port on the controller to the RS-232 data port on the codec [when using a Tandberg codec, for instance, you can connect a serial interface through Dataport 1] **and opening** an in-band data channel during the call. This eliminates ***most*** common network security concerns and provides directed communications that are sent via a dedicated channel.

Drawbacks:

1. Limited speed (most structures of this type will only operate at 38.8 kbps, for instance, well below the possible 100Mbps commonly available through an Ethernet interface).
2. Limited protocols (and, due to the limited bandwidth available, some protocols that are simply highly bandwidth intensive may not be ideal choices or, even, possible).
3. An 'older' form of communication, and may have some limitations when it comes to the breadth and depth of commands that are available. (Many systems developed today are migrating-to [and expanding their libraries of] XML-based commands over IP via Ethernet, and have frozen the command-libraries that are available over RS-232 Serial. Some systems have even eliminated altogether any RS-232 Serial connectivity).

Ethernet Communications with and between Control Systems:

Benefits:

1. In a properly designed packet network, this is reliable, stable and widely available.
2. Simple and inexpensive to physically connect.
3. Highly fault-tolerant at the transmission layer (TCP/IP is generally considered "reliable" or "guaranteed" delivery of information across an IP network).
4. Provides access to the current and next generation protocols for control scripts.
5. Generally provides much higher bandwidth transmission paths.
6. It is the path that is being followed by most Control System manufacturers.

Drawbacks:

1. Will, by nature, introduce random delays and variable packet loss in most feedback responses to initiated command sequences or requests for state/status data, and may not provide both rapid and reliable delivery of bi-directional data at great distances that is delivered in a timely enough manner as to properly invoke real-time responses to ad-hoc end-user changes in the environment and system state(s).
2. May create time-out and timing problems as a result of delay or 'jitter' in the network.
3. Susceptible to varying network load and latency conditions (Ethernet is a "shared" structure and commonly operates as CSMA/CD / CA/ CE).
4. May have some issues related to obtaining (or not being able to obtain) high-security approvals for attaching the Control System device to the organizations Ethernet network.
5. May have difficulty with Firewall and Proxy functions / elements in maintaining communication links and exchanging information with another controller or controllers.

Despite some very real challenges, in the face of the current evidence all effort should *most-likely* be made to provide primary communications with and between the Control Systems that are located at various sites via Ethernet, with a backup and failover or fallback redundancy to RS-232 Serial communications. As the new mechanisms of communication begin to fully take-over this environment, the dot-NET architecture, XML and Java coding, SIP and other new protocols that are now gaining momentum will provide advanced capabilities for Control Systems communications, especially with and between Control System frames, and these protocols are optimized for use in an Ethernet-based TCP and UDP / IP environments.

Conclusions: **This** paper (**Part 2 of 2**), **and** the **other papers and materials** referenced within, **are meant to provide a beginning foundation** for the exploration of broader and deeper utilization of the Control Systems that are already a critical component of the integrated videoconference systems, not only for the purpose of automating many of the functional elements of the system, but for the more advanced purpose of providing configuration states that are optimized for natural [and virtually "real"] human communications. By using the Control Systems Background Intelligence to work on an advanced basis, the "foreground" user-interface and end-user experience can be dramatically enhanced, in ways that were not previously possible through other mechanisms and means.

As we have already stated in some detail above: **This involves the consideration of many elements** and a few of these are listed for review below, **not** in any priority order. This list should serve to summarize and highlight certain specific points already covered within the body of this and preceding papers related to this topic.
1. The essential psycho-sociological dynamics of human-to-human communication must be considered within the context of the specific application activity for video communication within and by the end-user community. **This consideration** and understanding **must be a predicate to any selection of hardware or integration of functionality** within the installed systems.
2. The elements of technology that are selected for the integration must flow **from** these "human-to-human" dynamics. Higher performance flexible hardware will be, at times, ***required*** in order to fulfill the needs of the pre-defined and pre-determined set of human communication dynamics.
3. **The process of analysis for determining the complete solution is driven by** knowledge of and communication with the end-user community, understanding of the various applications for the communication enterprise, complete and thorough understanding of the level to which any component or set of components is able to perform in order to handle the demands placed upon it as an element of the total integrated system solution, and a process or method of constant feedback and adjustment for fine tuning the solution set as it is being developed and after it has been deployed.
4. If properly pursued, employing **Control System Background Intelligence should work to make the actual user interface to the systems simpler, not more complex**. That being said, **the user-interface** to the Control Systems **remains the most important element of the integrated design**, since it alone is the artificial electronic interaction point commonly accessed by all types and levels of end-users.
5. **Redundancy**, **Fail-Over**, **Durability** and **Reliability are all tied-together**, and the central thread that makes them all possible at the highest level of performance is the reasoned and full use of Control System Background Intelligence.

6. The use of **Control System Background Intelligence must be applied in such a way as to directly assist [not hinder] the Technical Operations and Support Staff** as they work to not only manage the lower levels of the enterprise (connecting calls, powering components, setting presets, performing system diagnostics and upgrades, answering real-time questions posed by the end-users), but also as they work to aid the end-users in facilitated and optimized experiences with visual video communication.

7. **Common-sense**, r**ealistic budgetary constraint**, **available technology** and the **level of sophistication of the user community** and their applications are all part of the evaluation and design process when it comes to first-class Control System Design and the application of Control System Background Intelligence.

8. The "**laws of intended and unintended consequence**" **will be heartily at-work** as the design team attempts to understand, develop, integrate and manage this type of advanced solution set incorporating atypical levels of Control System use. A change in any one element within the local system and any remote systems that will become connected at the Control System Intelligence layer will have an impact on one or more other elements within the designed solution set. The ability to see this effect and potential impact [both positive and negative] in advance, and to design with minimally invasive unintended negative consequences, is possibly the greatest area of challenge, second only to determining the hierarchy of elements to be served and serviced by this capability.

As always, we are pleased to offer this material to you for your consideration, and it is our hope that this will help expand the possibilities for the deployment of fault-tolerant, flexible, user-friendly and powerfully effective visual video communications systems.

Additionally, we encourage other design and integration specialists and professionals to document, catalog, write-up and share [publish] their own insights and experiences with others in this field. *Shared*-knowledge is power. The community of members associated with various industry groups [such as INFOCOMM.ORG] benefits each time one or more members provide opportunity for shared-insight and consideration by others.

Submitted by S. R. Sharer to INFOCOMM.ORG for exclusive publication and posting – August 2006.

CDG Inc. is a Design, Engineering and Business Communications Consulting Services Company specializing in all forms of video communications and videoconference in business, education and government.

Over 14 Years of Video Communication Design, Engineering & Business Consultation

For more about Scott Sharer, his credentials, background and his work,
you can right-click-on the graphic below in order to visit his on-line profile at: