# AoIP Delivers on Flexibility, Is Controllability Next?

The transportation of audio over IP is profoundly flexible, yet the ability to retain controllability over all these IP connected devices remains a frontier for discussion. This article considers approaches to matching the flexibility of AoIP network structure with new technologies and standards for control and monitoring user interfaces.

Photo 1: Markus Schmidt, DeusO GmbH co-founder, discusses the Microsoft Production Studios Cabasa prototype. The diagram outlines device control descriptor concepts and how web-based unified interfaces offer promising gains for controllability in flexible AoIP networks.

By
**Matt Hardy**
(DeusO GmbH)

Audio over IP (AoIP) has proven to be a transformative technology paradigm. As studios and broadcasters replace previous audio transport interfaces with IP-based solutions, they benefit from the IP network topology. The flexibility introduced through these network structures has inspired countless audio and broadcast professionals worldwide, with some characterizing AoIP as "profound" and "revolutionary."[1]

## Got Control? Field Trip to Redmond, WA

Recently, this topic was explored at Microsoft Production Studios. Late in 2016, lead engineer John L. Ball had completed the transition to AoIP utilizing equipment enabled by Audinate's Dante solution. Ball was looking for ways to more effectively wield the flexibility that the Dante solution had delivered. So he approached a likely bunch of companies, members of the Open Control Architecture (OCA) Alliance, asking, "You guys are working on interoperability, how can I get better interoperable control of our AoIP devices?"

Our initial introduction to the studio complex in Redmond, WA made us think about alternative ways to approach control of large networked AoIP systems and devices, utilizing the web browser as a platform to deliver unified control interfaces. In conjunction with Yamaha Corp. and with the Manager of Audio Product Development Division at the time, Hiroshi Hamamatsu, our team formulated a project plan for a fully functional prototype to prove our newly fueled R&D concepts.

This project was code-named Cabasa and a fully functional prototype was eventually delivered over a three-day workshop at the Microsoft campus in Redmond late 2017 (see **Photo 1**).

A number of the underlying features introduced through Cabasa are explored in the following sections. Academic ideas concerning complexity, flexibility and controllability, the importance of standards based technologies as well as UX and user-centered design all help formulate strategies for enhancing interoperability. These aspects are important for commercial imperatives intent on finding additional efficiencies in organized studios or broadcast operations.

## Exploring Complexity Theory, Flexibility, and Controllability

There are a number of perspectives on the topic of organizational complexity and flexibility, both with academic and practical interpretation. We start with exploring some academic perspectives, which provide some degree of parallel to the practical applications observed in the field.

This section is not a technical instruction and any concepts referred to here are not concerned with analysis of device performance or best architectural design processes, per se. It is more about how conclusions from an academic perspective support arguments for a new software design paradigm for user interfaces in workflows. The intent, therefore, relates to possible directions in design thinking, in conjunction with AoIP, where specific commercial goals exist to increase workflow flexibility in complex systems without disrupting controllability.

In the 2014 MIT paper, "Flexibility, Complexity, and Control of Large Scale Systems," David Broniatowski and Joel Moses discuss approaches to complexity theories, examining hierarchical tree structures and networks.

Moses outlines how hierarchical control is important to systems architecture because higher elements in a hierarchy have the capacity to control lower elements. He provides a definition of flexibility, where "a system can change its configuration in response to a change in the environment."[2]

Consider the tree structure as illustrated by Broniatowski and Moses in **Figure 1**. Pure tree structures are easy to control due to each node having exactly one parent. Layered hierarchies are defined by their nodes having multiple parents and the possibility of horizontal links. Networks are non-hierarchical and are highly flexible.

A pure tree structure provides clear paths and can be easily broken into sub-trees for purposeful functionality. An example might be an Access Control Model for a sub-branch (as proposed by Alban Gabillon, et al).[3] Where a system demands fine grain access control, there is a degree of attention required on what can be controlled and by which mechanism and through which method. So in this sense, fundamental concepts underlying control depend on hierarchical structures in which elements are ordered according to a relation of superiority and subordination.[2]

When examining the structures and general difficulties described by Moses, parallels appear with the similarities in how current audio device control software is organized into complex AoIP systems. For example, after witnessing the process Ball followed to configure various parts of his equipment, we discovered it involved multiple top level single entry point workflows (see **Figure 2**). The workflows resembled pure tree structures requiring systematic steps into each sub-organization of parts for each device connected to the larger system. This workflow highlighted an inherent focus on control by the existing installed software, leading to inflexibility, a quality of a pure tree structure Moses also describes.

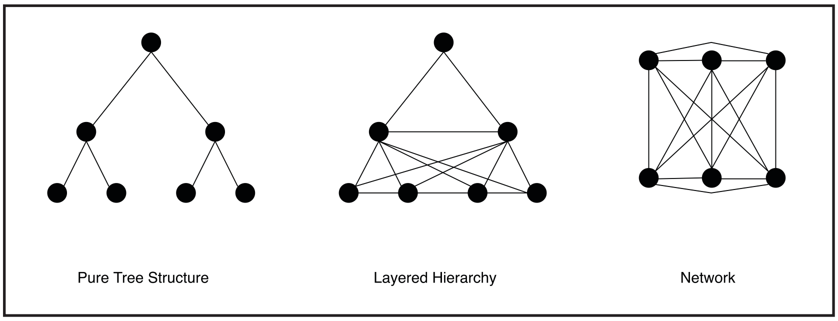

Figure 1: Organizational tree structure as illustrated by David Broniatowski and Joel Moses

It should be noted that to describe installed software as inflexible is not a negative criticism of that software, as it is an inherent design feature for that software's correct and proper operation. An apt heuristic of this dilemma is encapsulated by: "One person's architecture is another person's detail. One person's system is another's component."[4]

As with the larger system, each software had its own architect and each device its own set of components with finer details limiting a specific software's necessary awareness or ability to be interoperable with its peers. Consider an expansion of the problem Ball presented—how can the user control multiple parameters in multiple devices with a single point of entry?

One advantage of networked devices is they are already IP addressable. The potential then exists to expose each device as a tree of functionality (referred to as objects). One possibility of addressing devices, with a suitable control protocol, is that each object in a device could also be addressed and abstracted into an additional user interface, bypassing installed control software altogether. However, once all devices and functionality is openly available, the risk of losing controllability is increased due to the resulting networked topology of fully connected device objects. A network structure quickly becomes so complex, controllability in this paradigm will be reduced.

Moses established that hierarchy is essential to controllability, so a highly flexible but pure network approach will not help. In reference to layered hierarchy structures, Moses says "as the number

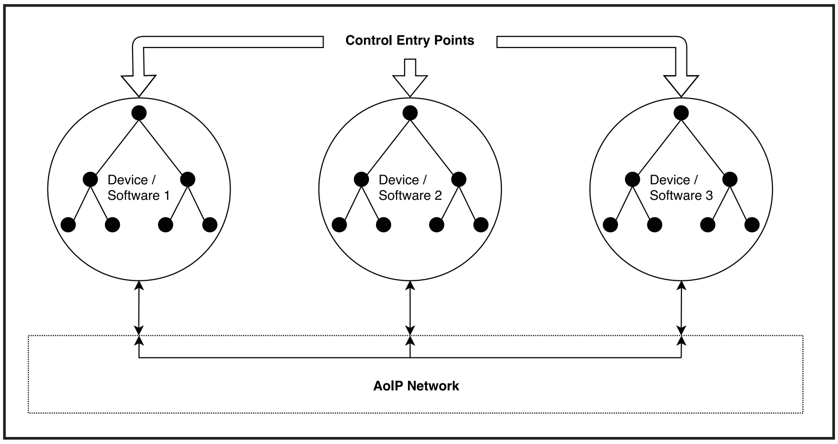

Figure 2: When examining current audio device control software organized into complex AoIP systems, we discovered multiple top-level single-entry-point workflows.

of horizontal links and multi-parent vertical links grow, either in human organizations or engineering systems, the flexibility usually tends to develop while keeping the increase in complexity in check." Key to this statement for practical interpretation is not that a user will necessarily want lateral connections or single points of entry for every node or branch to give them more "flexibility" (see **Note 1**). The requirement here is to understand what it is that a user additionally wants to achieve from a workflow perspective, and whether a system is flexible enough to achieve the stated goals without costing controllability.

Photo 2: Cisco routers at Microsoft Production Studios power the audio-over-IP installation. In the approach taken for the Cabasa project, it was possible to detach the user's need to think about a network at all. The UI incorporated a filtered list of channels able to be connected for each audio channel available, using a select/connect widget.

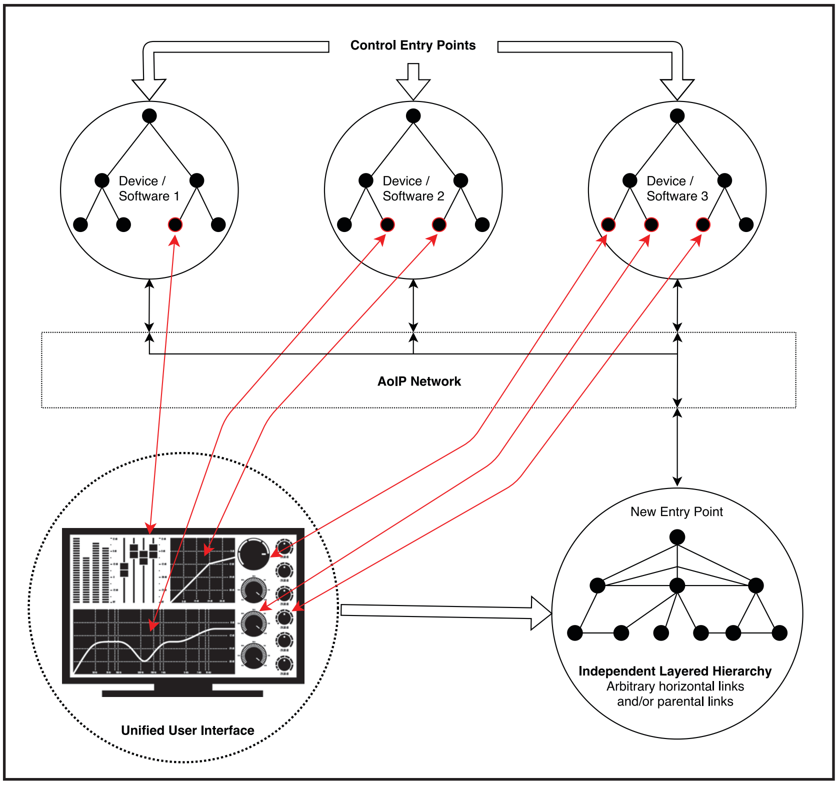

Figure 3: The Cabasa prototype can create a new structure, in parallel to existing control system structures, with the possibility for multiple parents or lateral connections

## Workflow Demands Drive Change

Ball provided our team with many critical perspectives and his concerns were not just about saving on operational time. Ball noted that, "at Production Studios, because multiple people access our Audio over IP devices, being able to secure this access to specific functionality from a single workflow UI [web browser] would be a huge help for us."

Case studies exist for why workflow drives the pursuit of flexibility as observed at Redmond, and at other installations. All revolve around the reality of multiple users with differentiation in roles. The differences involve automation options, differing equipment requirements, space or location, security clearance, and it's always possible requirements are found as new projects unfold.

Blair Liikala, media engineer at the North Texas College of Music, revealed in our discussions on this topic that a "problem with web-based UI has been around presets, and managing multiple devices along with keeping things in sync." This observation focuses on how roles and workflows are influenced by user demand and existing control software realities. Blair suggests that choosing a web browser for control is not the central issue, rather it is about how roles and workflows define system boundaries. The web browser platform can support the industry in moving toward a user-centered control software design, but it is how underlying system structures are implemented that remains of higher importance.

Flexibility in a system is key to fulfilling the types of demands encountered (see **Photo 2**). Liikala explained, "For all our production computers I simply have the browsers logged into an account that syncs bookmarks and that bookmark shows up on every machine. When an update for the web-UI comes out, I only have to update the device, not every single computer." With a contrasting perspective, Ball described "regularly receiving calls from talent asking [John] to turn down one of the producers mic gain due to issues with level/voice power."

The common theme for both Ball and Liikala is they want the ability to more easily access a given workflow within their respective network topologies, without the risk of disruption. Also, at any point, a user may need the system to function in ways that the installed control software may not have been designed to do. These points of view are reinforced by simply contrasting user roles (i.e., Admin vs. User or Producer vs. Artist).

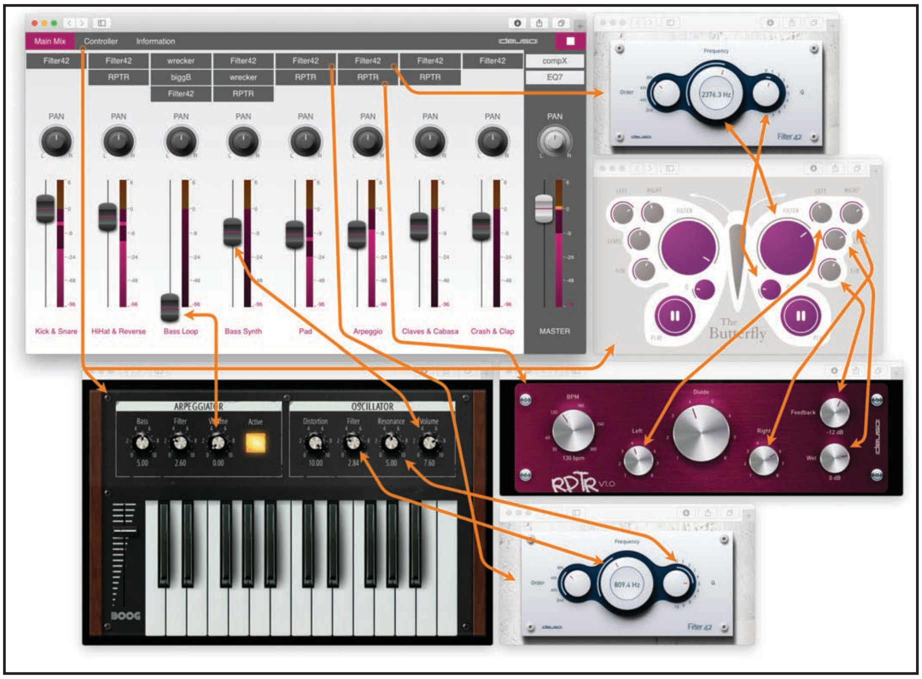

Photo 3: This collection of user interfaces illustrates custom UIs of similar collections of functionality. These functional widgets can be arbitrarily combined and recombined, with various web-based user interface examples for plug-in DSPs, instruments, and control surfaces. Arrow links indicate which widgets are unified in alternate UIs, utilizing layered hierarchial structures.

The situations described by Ball and Liikala relate to the same question: Can system control workflows reflect the needs of the user, or is the user forced to adapt to the needs of the system control software?

## Building Workflow Hierarchies in Network Structures

Independent layered hierarchies offer a way to meet requirements in flexible ways, and not disrupt controllability, but also do not necessarily require any changes to existing system control software.

The Cabasa prototype approached the workflow problem by building independent layered hierarchies where lateral connections and multiple parents could be bound directly to the objects throughout the network of devices (see **Photo 3**). A key strength of this approach is its ability to maintain hierarchical controllability using hierarchical structures, while delivering great flexibility through the possibility of unlimited lateral connections.

**Figure 3** illustrates how the Cabasa prototype creates a new structure, in parallel to existing control system structures, with the possibility for multiple parents or lateral connections. This approach enabled Cabasa to re-establish the primacy of user centered design for control user interfaces and workflow simplification.

In addition, building on a web browser platform resulted in no disruption to the existing control software paradigm, as IP native web browsers exist happily on any modern desktop, laptop, tablet, or phone.

Cabasa demonstrated unified user interfaces. They involved gains, level knobs, faders, connecting Dante transmitters and receiver by channel, device metering, phantom power and EQs all on web-based user interfaces (see **Photo 4**). These interfaces controlled multiple parts of multiple devices, cross-vendor and even over different studios located throughout the 65,000 ft$^2$ complex (see **Photo 5**).

## Road Maps Involved with Interoperable Control

With workflow as a driver of change, the user role forms a central focus, as it is the user (producer, staff, or performer) who commands significant economic value in the context of live productions or other studio activities.

Workflow efficiencies are, therefore, tightly bound to a system's flexibility when the demand on system configurations change. With these various considerations let's look at technical strategies to deliver both flexible control and functionality, which adapt to systems prone to complex configuration changes.

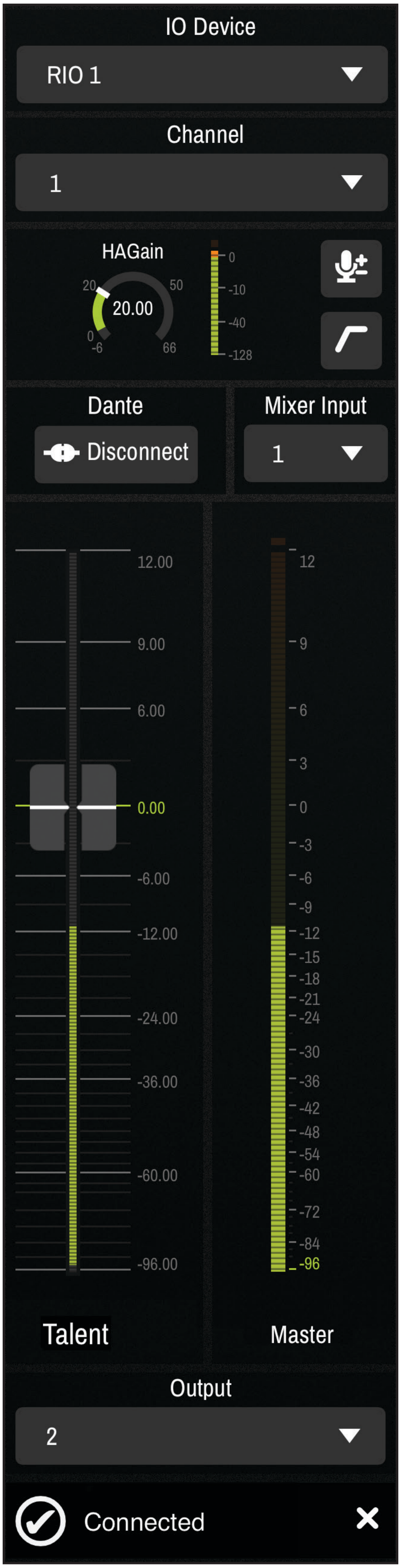
IO Device
RIO 1
Channel
1
HAGain 20.00
Dante: Disconnect
Mixer Input: 1
Talent
Master
Output
2
Connected

Photo 4: An example of a Yamaha CL-3 console channel UI incorporating a Rio Series gain with a 48v toggle, connecting pre-filtered available Dante channels.

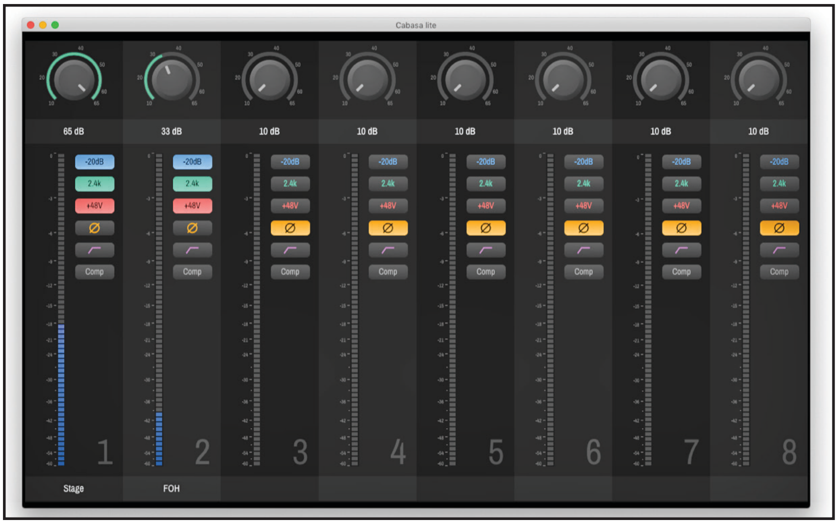

Photo 5: A third-party web browser UI controls all the AES70 objects in a Focusrite MP8R (an eight-channel remote-controlled mic preamp and A-D interface for Dante AoIP networks). It would be possible to unify each strip with additional parameters (e.g., a console channel) and utilize both over Dante/AES67 + AES70.

## Web Browsers, Ready When You Are

Web browsers have typically not been considered high-performance software, and perhaps even relegated to a "poor cousin" of the natively compiled C++ and Objective-C applications that users have relied on over the last decade. However, the game has changed.

The web platform is now critical in many sectors, from banking to Citizen-Government communications. Critical to this was better performance, robustness, and continuity. The web browser now offers a reliable platform for future-proofed web technology strategies.

Arguably, standards are what provides the web browser platform with the continuity commercial operations need. Leaders in the standards-based web browser development use open-source code, rarely break backward compatibility and have driven performance to rival natively compiled applications (e.g., Chrome delivered 20× performance increases between 2008 and 2018, see 10 years of speed[5]).

## Standards-Based Control Protocol

IP network control protocols are fundamental to building audio device control software applications. From a practical controllability perspective, while devices commonly have some form of control protocol onboard, it's common that not all share one standards-based protocol. A standards-based control protocol delivers continuity and commonality, enabling interoperability to become a natural expectation for workflow architects.

OCA is a standards-based control protocol specifically for media networks. OCA or AES70, as it is also known from the AES standards ratification, offers a modular approach to enabling control for many different types of network devices. Because devices may require user interfaces across multiple control software or vendor installations, device interoperability is seen as an important part of the OCA vision.

Consider the concept of a layered hierarchy described in the Cabasa prototype, each device has an object tree of controllable parameters, OCA/AES70 can be used to expose these as controllable objects over a network topology (see **Photo 5**). Each individual OCA object contains type information and all its functional parameters. For example, a gain might have minimum, maximum, and step information, an EQ an even greater number of controllable parameters. Cabasa could take any device with AES70 support, discover its object tree, then generate a user interface matching each object's set of functionality, thereby providing a user unlimited parameter unification possibilities. High levels of controllability can be retained by the independent layered hierarchy.

The business case of a standards control protocol exists. First, the value of continuity from a standards-based protocol is retained after the initial implementation investment. Second, more interoperable network design possibilities can be realized. Should all devices naturally expose controllable parameters with AES70, integrators and workflow designers can choose OEM workflow, control, and monitoring technologies that do not restrict them to only the specific vendor's software or software platforms.

## UX Design

The importance of user experience (UX) in designing workflows is relevant in the vision of layered hierarchy structures. When following a user-centered

## Notes

1. David Broniatowski and Joel Moses produce evidence that supports the statement that "flexibility of layered hierarchies comes both from having multiple parents and having several horizontal links; however, adding horizontal links tends to increase flexibility much more quickly than does adding multiple parents. In contrast, only horizontal links greatly disrupt controllability because they necessarily involve the addition of loops."[2] It is the lateral connections which need to be carefully considered when designing an augmentation or transition of an existing pure tree to a layered hierarchy.

## About the Author

Matt Hardy is Business Development Manager at DeusO GmbH. He has 20 years experience with web-based technologies and organization development using the web platform. Matt holds a bachelor degree in Philosophy & Management and is currently focused on organizational development and strategy.

UX design methodology, users tend to provide qualitative evidence for how they specifically find operational efficiencies. It follows that the flexibility in a control software comes from avoiding rigid structures. How to transition from pure structured control softwares to solutions that resemble layered hierarchies is therefore a value consideration driven by user demands for flexibility and efficiency gains. The value proposition of utilizing both a standards-based control protocol and standards-based user interface platform is a specific ROI in which a UX Design process can assist in developing.

## Complexity is Required, Allow Simplicity to Reign

Having considered how complexity theories help underpin the practical problems facing system integrators seeking more flexibility, the primacy of user experience from a control perspective is reinforced. This discussion involved various problems faced by network engineers, differing paradigms, practical examples and possible technological road maps.

Through combining modern web technologies, communication standards and more flexible hierarchy structures, future users of control interfaces may not even realize the scale of complexities underpinning the potential of much simpler customized workflows.

In the case of independent layered hierarchies, it is possible to make workflows appear to the user as a collected simplification of essential functionality, they simply get on with more efficient workflows. Having a user not realize how complex a system is or understand these complexities, is indeed part of the point. Encapsulate the approach to unified user interfaces with another commonly used heuristic, "Simply, simplify, simplify!"

*Author Acknowledgements: I would like to thank the following contributors. John L. Ball, Microsoft Production Studios; Blair Liikala, North Texas College of Music; and the support I received from Hiroshi Hamamatsu, Markus Schmidt, Arne Gödeke, Tobias Josefowitz, Jeff Berryman, and Douglas Owens.*

### References

[1] P. Walker, "Audio Over IP - Why It Is Essential," Calrec Audio, 2018. www.thebroadcastbridge.com/content/entry/10456/audio-over-ip-why-it-is-essential

[2] D.A. Broniatowski and J. Moses, "Flexibility, Complexity, and Controllability in Large Scale Systems," CESUN 4th International Engineering Systems Symposium, June 2014.

[3] A. Gabillon, M. Munier, B. Jean-Jacques, L. Gallon, and E. Bruno, "An Access Control Model for Tree Data Structures. In: Chan A.H., Gligor V. (eds) Information Security. ISC 2002. Lecture Notes in Computer Science, Vol. 2433. Springer, Berlin, Heidelberg

[4] R. Spinrad, *The Art of Systems Architecting* (2nd Edition). CRC Press, Inc., 1989, as cited in Mark W. Maier and Eberhardt Rechtin, 2000.

[5] A. Osmani, "10 Years of Speed in Chrome," Chrome Blog, September 2018, https://blog.chromium.org/2018/09/10-years-of-speed-in-chrome_11.html