Audio Engineering Society

Convention e-Brief

Presented at the 133rd Convention
2012 October 26–29 San Francisco, CA, USA

Modeling Auditory-Haptic Interface Cues from an Analog Multi-line Telephone

Durand R. Begault[1], Mark R. Anderson[2], and Rachel M. Bittner[3]

[1] Human Systems Integration Division, NASA Ames Research Center, Moffett Field, CA 94035 USA
Durand.R.Begault@nasa.gov

[2] Dell Systems, NASA Ames Research Center, Moffett Field, CA 94035 USA
Mark.R.Anderson@nasa.gov

[3] Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 USA
rmb456@nyu.edu

ABSTRACT

The Western Electric Company produced a multi-line telephone during the 1940s-1970s using a six-button interface design that provided robust tactile, haptic and auditory cues regarding the "state" of the communication system. This multi-line telephone was used as a model for a trade study comparison of two interfaces: a touchscreen interface (iPad) versus a pressure-sensitive strain gauge button interface (Phidget USB interface controllers). The experiment and its results are detailed in the authors' AES 133rd convention paper "Multimodal Information Management: Evaluation of Auditory and Haptic Cues for NextGen Communication Displays". This Engineering Brief describes how the interface logic, visual indications, and auditory cues of the original telephone were synthesized using MAX/MSP, including the logic for line selection, line hold, and priority line activation.

1. REFERENCE DESIGN

A ubiquitous telephone design, seen in models produced by the Western Electric Company (a subsidiary of AT&T) during the 1940s-1970s, used a multi-button interface design for managing incoming calls across different "lines" (hold button, line selection, priority line selection). The basic design, exemplified in many variants, remains influential for modern office desktop telephones. Its simplicity exemplified a successful human factors approach to the problem of managing multiple lines of communication. It featured raised mechanical buttons that provided robust tactile, haptic and auditory cues, unlike the touchscreen or membrane switches used in modern equivalents.

Figure 1 shows a six-button interface of the Western Electric 565 telephone. The buttons are mechanical two-state switches that provide several forms of haptic, auditory and visual cues as to their current status. The spacing and layout of the buttons conforms to the hand and fingers for ease of operation. Due to their nearly 0.5-inch raised profile, tactile exploration the switch set without "consequence" (accidentally engaging the button) is possible using only tactile and auditory cues. The switches make a sound when touched lightly (without engaging). Pushing the switch in fully makes a distinct two-part "engagement" sound (one at the end of the throw, and one after the release). The "hold" button has its own sound characteristics and causes the currently engaged button to be released, resulting in yet another auditory cue. While no published research relating to these line selection switches has been found by the authors, Deininger [1] describes research into the relationship between force-displacement of telephone number keys and user speed, accuracy and preference; Figure 2 shows an adjustable "universal push-button switch" used at Bell Laboratories.

Figure 1. Western Electric Model 565 telephone.

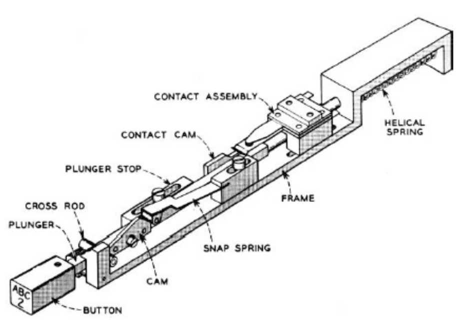

Figure 2. The adjustable "universal push-button switch."

In addition to haptic and visual cues, the telephone uses two audio alerts. Most often, the familiar classic telephone ring is used to direct visual attention to the telephone receiver so that the line associated with an incoming call can be answered. The audio alert is unspecific as to the incoming line (counting from left to right, buttons 2-5). However a second sound, descriptively termed a "buzzer," indicates that a call has come internally via the intercom, and is answered using the rightmost button (button 6). Hearing the classic telephone ring compels a subsequent action for first visually identifying the appropriate flashing button (button 2, 3, 4 or 5), and then engaging it by pushing the button inward (a "visually-guided haptic target").

2. EXPERIMENT SOFTWARE OVERVIEW

This multi-line telephone was used as a model for a trade study comparison of two communication interfaces proposed for future flight decks. Human performance and preference were measured in several ways, primarily by comparing a touchscreen interface (iPad) versus a pressure-sensitive strain gauge button interface (Phidgets, www.phidgets.com) under "audio feedback" versus "silent" conditions. The interfaces, experiment and results are detailed in the authors' companion AES 133rd convention paper, "Multimodal Information Management: Evaluation of Auditory and Haptic Cues for NextGen Communication Displays." This engineering brief and accompanying poster presentation describes how the interface logic, visual indications, and auditory cues of the original telephone were synthesized using Cycling74's MAX/MSP version 5 software, including the logic for line selection, line hold, and "priority."

The logic patches of the software include means for assigning and buffering incoming messages, depending on the current state of the device. Patches specific to the use of the touchscreen or Phidget are selected depending on the experimental condition.

The final module in the MAX/MSP pipeline models the current state of the array of switches using multiple logical operators. Multi-line telephones allow only one line at a time to be audited, and engaging one line would "hang up" other lines unless they were first put on "hold" using a hold button. Because the participant may elect to handle the messages in any order, this module assigns the message to the lowest open line, or buffers the message if no line is available.

3. DETAIL OF THE ALGORITHM

The custom experimental control software BAM (Button Audio Manager) is composed of a hierarchically arranged series of MAX patches that model the telephone's visual and auditory cues, run the experimental blocks, and collect the data for subsequent analyses. Figure 3 shows the highest level of the experimental control software. For each experimental block, the encapsulated object "Brain_v3" reads a script of time-ordered events that includes lists of audio messages, audio alerts, visual display changes, and user response questions that are displayed on a second iPad. All user actions with external interface hardware are recorded for subsequent analysis. Lower levels of the experimental control software are devoted to simulating the telephone: producing button sounds in reaction to touch, providing visual feedback in the form of flash rate and illumination, and indicating the state logic of the buttons (line engaged, disengaged, held, or pending).

Analysis of the original telephone indicates a complex set of "cause and effect" interactions that the software is required to simulate using the Phidget or iPad interface. For example, to "hold" line 1 and "answer" a new incoming call requires the software to accomplish the following tasks:

- The object "Assigner" (circled in the lower right of Figure 3) finds the next available line to assign to the incoming call (ex. if line 2 was already on hold, go to line 3)'. "Assigner" then sends a message to the object "Sound Maker" to activate ring tone (upper right circle, Figure 3) and to the object "Led State" to flash the light associated with the incoming line button.

- When the user pushes the hold button, the object "Button Logic":
    1) activates the hold button audio and visual cue (red light)
    2) pauses the message associated with the current activated line (via a command routed to the appropriate object "MsgPlayer")
    3) makes the line 1 light go to a rapid flash
    4) activates the audio cue corresponding to the release of the line button (activating the "onebang" command to the object "send SwInA" when exceeding a threshold of "500" from the hold button on the interface).

- When the user selects the incoming line button, the object "button logic" sends messages to the appropriate objects to activate the audio and visual cues associated with the button push (change the visual cue from a flashing light to a constant light; play back the new message).

The buttons of the Western Electric telephone were sampled for playback of five unique sounds: a relatively quiet "plastic rattle" corresponding to lightly touching the button; two sounds for button engage (corresponding to the application and release of finger force); and two sounds for button release. A constant stream of integers from 0-1000 is sent from each strain gauge button of the Phidget, depending on finger pressure. The iPad sends data based on duration of touch within the area of the button, scaled to 0-1000 via a custom application resident on the iPad. The integer values are used by the software to determine, for each button, when they are being touched, pushed to engage, or pushed to release.

The object "PhidgetBtnMan" in Figure 3 examines the state of each button. The first two arguments "600 450" refers to: when the finger pressure (or timing for the iPad) button exceeds 600, activate the first auditory cue (audio sample of application of finger force for button engagement), and activate the second auditory cue when the finger force is released from the button, corresponding to the threshold dropping from above 600 to below 450. The third and fourth arguments "600 200" activate the third and fourth auditory cues that correspond to the button release sounds. The equivalent data from the iPad corresponded to finger timing within a specific area. Figure 4 left and right detail the objects "PhidgetSmall" and "iPadControl" in the upper left of Figure 3. These objects control how messages are received and routed from the strain gauge device and the iPad, respectively. The outputs 1-6 are sent to the five objects shown in Figure 3 as "PhidgetBtnMan."

This work was supported by NASA's System-wide Safety Assurance Technologies (SSAT) Project.

4. REFERENCES

[1] R. L. Deininger, "Human Factors Engineering Studies of the Design and Use of Pushbutton Telephone Sets," *Bell System Technical Journal* vol. 39, pp. 995-1012 (1960).

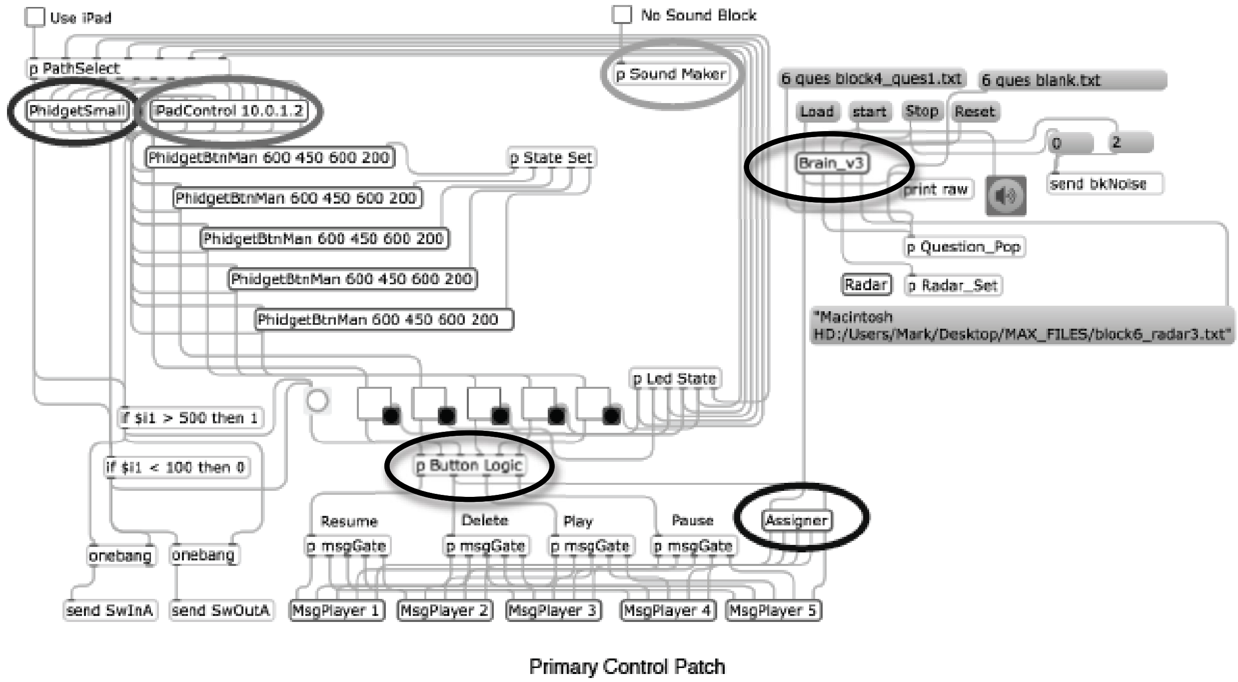

Figure 3. High-level MAX patch for BAM software.

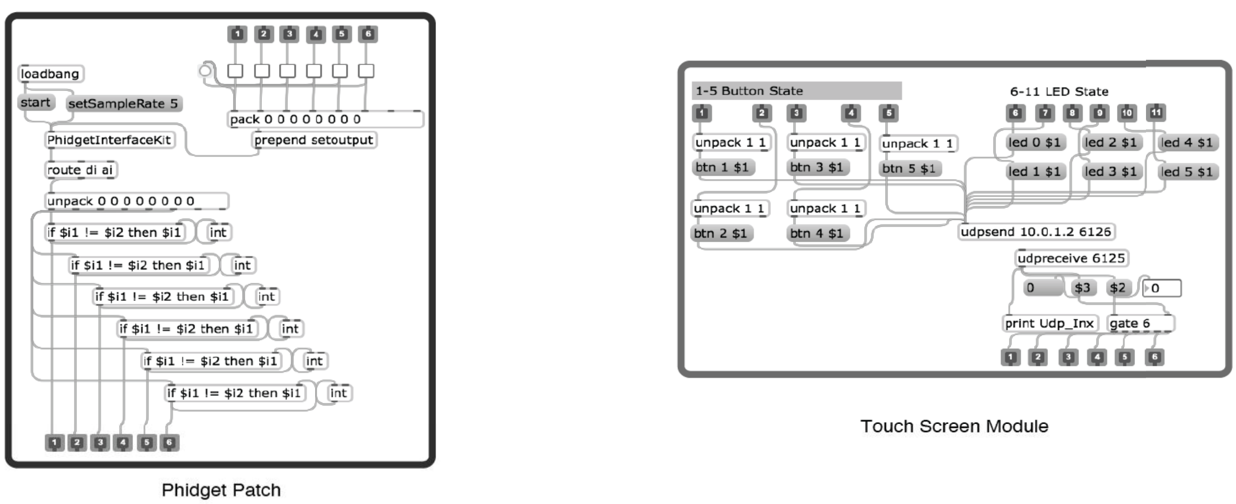

Figure 4. Details of the objects "PhidgetSmall" and "IpadControl."