Usability Evaluation

jEAC
Real-time 2D/3D EAC Interaction

Andrew Kipfer
Ryan R. Varick

April, 2006

Executive Summary

This document outlines our evaluation of analysis of jEAC, an application designed to facilitate interaction with the Extended Analog Computer (EAC) (see Appendix A). Our evaluation focused primarily on task-based analysis of common scenarios performed by expert users. We designed our test to identify possible usability issues to be addressed in future development.

Test Population

We invited five (5) computer science students to participate in a lab-based evaluation session. All participants were familiar with analog computing and representative of the population for which jEAC was designed. All participants were selected from Prof. Jonathan Mills' VLSI Design class, and were offered the option of extra credit for their participation. Participants were asked to sign an informed consent statement, as a courtesy.

Test Procedure

Participants were tested individually in a laboratory setting. Each participant was asked to complete two structured tasks, vocalizing their reactions at each step. The first task was to configure the analog computer to a know configuration. This test was intended to test the basic connection and manipulation controls. The second task tested participants' applied knowledge of analog, using the interface controls to perform a short computation.

Recommendations

Major recommendations include:

- Better label available EACs, differentiate between types.
- Ask the user to attach uEACs before running auto-probe code.
- Add the LLA function to the Inspector window.
- Add the LLA ID to the node map button.
- Open support windows outside of the main application window.

Table of Contents

Introduction

This document describes the motivation, method, and results of our evaluation of jEAC. Our tests were designed to identify areas of improvement for future revisions of the application. To this end, we devised a two-part test: part one explored the basic operation and layout of the application while part two concentrated on a subset of advanced features.

Application Description

jEAC is pedagogical application designed to help researchers better understand the mechanics of the extended analog computer (EAC). It integrates configuration of the EAC with real-time output visualization along three axes: 3D voltage gradient visualization, 2D voltage gradient visualization, and timed LLA polling. jEAC is believed to be the only application of its kind in the analog computing field. Further information is attached under Appendix A.

Rationale

As with any novel interface, it is important to evaluate the effectiveness of the underlying interaction method. For jEAC, we decided to focus first on identifying major usability issues. MORE

Population

Users are expected to be either computer science students in Prof. Jonathan Mills' courses, or researchers in related fields. All users are expected to be technically literate; advanced users by traditional measures. For the evaluation, we selected five students from Prof. Mills VLSI Design course to participate in individual, lab-based testing. Four males and one female were chosen; all were given the option of extra credit for their participation.

Evaluation Objectives

Our tests were designed to collect data in three areas:

1. Interface – test the overall interface design, including placement of controls and labeling.
2. Representation – test the underlying interface metaphor (EAC controls and visualization).
3. Utility – gauge the utility of the application (i.e., "can participants use it?")

Methodology

Given our evaluation objectives, we created a two-part test for participants to attempt. The first was a straightforward, mechanical task designed to evaluate the objectives [1] and [2]: overall application layout, control selection, and interface metaphor (graph orientation). The second task was applied, and asked users to interact with the application to complete a simple truth table (objective [3]).

Refer to Appendix B for the actual testing materials administered.

Task A – Configure the EAC

The first task asked the participant to configure the analog computer to a known state. This task was broken down into five distinct steps:

1. Open jEAC.
2. Connect to the EAC.
3. Reset the EAC.
4. Attach and configure two sources.
5. Attach and configure one LLA.

Task B – Verify the configuration

The second task was stated less explicitly. It asked users to use the configuration from the previous task to construct the truth table for analog-XOR, a known problem in analog. We selected participants that were familiar with this problem. While not made explicit, an expected sequence of actions was:

1. Open the LLA Inspector.
2. Modify source configurations.
3. Read the output of the LLA.
4. Repeat.

Results

Findings are grouped by objective and given a severity (High, Moderate, Low, or Informational):

Interface

ID	Encountered	Users	Severity	Description
Driver labels	Task A	All users	High	Users found the choice of EACs confusing. Most first tried to connect to a network EAC (eac#.cs.indiana.edu) rather than the uEAC on COM#.
Latency	Both tasks	Some users	Moderate	Some users noticed that the application was slow to respond when performing some operations, such as adding or removing a component.
Connection	Task A	Some users	Moderate	Some users loaded jEAC before connecting the uEAC to the computer. Thus the uEAC was not found by the auto-probe routine.
Reset	Task A	Some users	Low	Some users seemed confused by the labeling of the reset command.
Controls (1)	Task A	All users	Low	Users used the text entry box to set the sources to 115 uA. Most hit ENTER to confirm the changes, though some were not clear if simple leaving the text field would apply the changes.
Control Persistence	Both tasks	Some users	Low	It was not apparent to some users that the Control windows could remain open after use.

Representation

ID	Encountered	Users	Severity	Description
Node controls	Task A	All users	Informational	Users employed different strategies to make sense of the node map. One experimented by adding a connection at (1,1), one initially tried to use the 3D graph, while others thought about the interface before proceeding.
Node labels	Task A	Some users	Informational	It was not immediately apparent to some users what the MxN node map labels meant.
Graph Interaction	Task B	Some users	Informational	When looking for the LLA output values, some users naturally discovered that the 3D plot could be rotated.
Keyboard Accelerators	Task B	Some users	Informational	One user noted the menu shortcut labels for Task B, while another explicitly requested keyboard support.

Utility

ID	Encountered	Users	Severity	Description
LLA Persistence	Task B	Some users	Moderate	Some users were not aware that the LLA Inspector could remain open. They entered a change-open-read-close loop.
LLA Output	Task B	Some users	Moderate	It was not immediately apparent to most users where to look for LLA output values. Some used the graph values; some looked in the Control window. Even after discovering the LLA Inspector, its use was not immediately obvious.
LLA Identification	Both tasks	N/A	Informational	While not directly noted by users, it is clear that there is no visual connection between an LLA on the node map and an LLA in the Inspector.
Controls (2)	Task B	Some users	Informational	In contrast to Controls (1), most users used the slider to reduce sources to 0 uA.

Recommendations

For Task A, each recommendation is tied to a specific step; for Task B, recommendations are listed in order of decreasing severity:

Task A – Configure the EAC

Step 1: Open jEAC.
- Connection: Ask the user to attach any uEACs before running the auto-probe code

Step 2: Connect to the EAC.
- Labels: Improve the labeling of available EACs, differentiate between types

Step 3: Reset the EAC.
- Reset: Change “reset all connections” to “reset EAC”

Step 4: Attach and configure two sources.
- Controls (1): Apply changes when text box loses focus
- Control Persistence: Open the Control windows outside the main application window
- Latency: Investigate ways to improve responsiveness of control UI
- Latency: Investigate ways to add feedback when modifying components in the Control window

Step 5: Attach and configure one LLA.
- LLA Identification: Add the LLA identifier to the node map label.

Task B – Verify the configuration

- LLA Persistence: Open the LLA Inspector outside to the main application window
- LLA Output: Add the function to the Inspector window.
- LLA Output: Add the input and output values to the Control window

Appendix A – Requirements Specification

Attached separately.

Appendix B – Usability Test Instructions

Your task is configure the uEAC to solve exclusive-or (XOR). This configuration has useful theoretical properties ï do not worry if you are unsure how to do this.

The configuration for XOR is known to require three (3) *components*: two current sources, and one Lukasciewicz Logic Array (LLA).

Part I ï Configure the uEAC.

1. Open jEAC, the interface for the extended analog computer.
2. Connect to the uEAC.
3. Verify the connection by resetting the uEAC.
4. Attach one of the sources to each the left and right edges. Set their initial current to 115 uA.
5. Attach one LLA in the center. Set its function to #18.

Part II ï Verify configuration.

Recall the truth table for XOR:

	0	**1**
0	0	1
1	1	0

Binary values (0 and 1) are not applicable to analog. We need to construct the truth table for analog-XOR. For each case, update the current on the sources and read the updated value of the LLA to complete the truth table:

1. Digital: (**1** xor **1**) = Analog: (**115 uA** xor **115 uA**) =
2. Digital: (**0** xor **0**) = Analog: (**0 uA** xor **0 uA**) =
3. Digital: (**1** xor **0**) = Analog: (**115 uA** xor **0 uA**) =
4. Digital: (**0** xor **1**) = Analog: (**0 uA** xor **115 uA**) =