

CAPSTONE PROPOSAL

THE HCI OF ANALOG COMPUTATION

Ryan R. Varick

Human-Computer Interaction Design

School of Informatics

Indiana University, Bloomington

October, 2005

Introduction

At its simplest, analog computing may be thought of as the opposite of digital. More precisely, they are two ends of the same spectrum. Analog is continuous computation, digital is discrete. Both have their strengths and weaknesses; each is suited to different applications. Analog and digital compliment one another. However, digital rules the day.

Why is this?

Digital computation owes its success to the analog computers of the mid-twentieth century. Much of what we think of today when we refer to “information technology” arose as a response to the difficulty of using analog computers. Traditionally, computing with analog really meant computing with differential equations. While powerful, the simpler rules of digital were easier to use. At first glance, punchcards may not seem to be much of an improvement over complicated mathematics; however, *abstraction* is the key. Punchcards soon gave way to simple electronic programming, which in turn led to higher-level languages with ever-growing expressive power. Soon, digital computers could sufficiently approximate problems that had previously only been possible with analog computers. The rest, as they say, is history...

Forty years later, we find ourselves in a similar situation. Digital is routinely asked to address problems that are increasingly difficult to quantify algorithmically. The resultant toolkits are so complex that they are inaccessible to all but the most gifted computer scientists. A more balanced approach is necessary, as evidenced by the renewed interests in “non traditional” computation. Research underway at Indiana University is exploring a new, more powerful analog paradigm with possible immediate applications in areas such as signal processing, network routing, pattern recognition, feedback control, and computer graphics.

Background

My interest in analog is long-standing. As an undergraduate, I took two courses with Prof. Jonathan Mills: *B443 Computer Architecture*, and *B644 VLSI*. These courses introduced me to the concept of the extended analog computer. As a graduate student, I took a course taught by Prof. Larry Yaeger, *I590 Artificial Life as an Approach to Artificial Intelligence*. In that class, I began to get a sense of a possible project analog application – an analog-based artificial life simulation. This led to an ongoing independent study, undertaken this summer.

My work to this point has focused on developing the necessary tools to enable my artificial life work. This is inherently an HCI problem: few people want to concern themselves with multivariate calculus and electrical engineering to solve a problem. For analog to prove itself useful to the computer science community, it must be easier to use. I am interested in exploring the HCI and interaction design issues associated with an emerging computational paradigm.

User Groups

Tentatively, I have identified two primary user groups:

1. Novice users, or *learners*, who are new to analog,
2. Experienced users, or *developers*, who are interested in applying analog.

These two groups have distinct needs. For novice users, there is a need for established methods to teach the fundamentals of analog and the extended analog computer. Developers, on the other hand, require robust, well-defined tools to use to solve problems.

Capstone Goals

My primary goal for the capstone year is to advance work that is of use to the analog community. At this point, my interests are broad, and I would like to contribute in many areas. However, it would be foolish to assume that one year is enough time to address all my interests. I therefore consider my *primary* obligation to be novice users.

With that said, my independent study is rather technical. The insights from this project are useful primarily to developers. To support my analog/artificial life research, I developed a toolkit that I hoped would be useful to the students of Prof. Mills VLSI class. Initial results have been disappointing – students have not had much success using my code. From this I have learned my first major lesson of the capstone process: assumptions should not be treated as fact! To ensure that my work meets *real, user-centered* needs, I need to employ the techniques of a human-centered design process.

Ideas

My first objective is complete my independent study. I would like observe and enumerate the needs of Prof. Mills' students, and other analog practitioners before deciding exactly what to focus on. Ideas that I am considering include:

1. *Further toolkit development.* Currently, I am trying to cater to too many needs with my toolkit. I believe there is value in robust, structured code, and would like to continue development of the toolkit.
2. *Develop an exploratory interface for novices.* While the toolkit would be focused on developer needs, a graphical interface to the extended analog computers may be useful for those are new to analog computing.
3. *Collect example applications.* In both cases, examples are useful teaching tools. I would like to collect and package existing analog applications in an easy-to-access, easy-to-run format.
4. *Create a research repository.* For researchers (a possible third, distinct user group), it is necessary to review necessary literature. A central repository of analog-related material would facilitate research interests.
5. *Create a video prototype.* Explaining analog to people is difficult. A video prototype may be helpful.
6. *Investigate an analog-based Phidget system.* Phidgets are modular components used to build pervasive systems. Physical sensors, attached to a USB-controller, generate input that is manipulated by client-side software. The same controller may be used to control physical servos and other output devices. I would like to explore ways that analog computers may be used to control phidget systems.

Timeline

This tentative timeline outlines possible capstone strategies:

October 15-November 15 (semester I):

- Review literature*, including major works, student papers, and related documents.
- Identify appropriate research techniques*, including contextual inquiry, ethnographic observation, task analysis, and video prototyping
- Begin HSC application process*
- Complete independent study*

November 15-December 15:

- Execute initial research agenda*
- Revise developer toolkit*
- Explore novice interfaces*

December 15-January 15:

- Begin novice interface development*

Post January 15 (semester II):

- Prototype and evaluate novice interface*
- Evaluate developer toolkit*
- Prepare relevant technical documentation and papers*
- Prepare capstone presentation and poster*

Sign-offs

Abandon all ye hope.

Student:

Primary Advisor:

Capstone Instructor:

Program Director:

Graduate Recorder:
