Process Makes Perfect

Marc Schwartz
B.F.A. Industrial Design
Rhode Island School of Design
June 2000

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
In partial fulfillment of the requirements for the degree
of **Master of Science in Media Arts and Sciences** at the
Massachusetts Institute of Technology
June 2005

Author: **Marc Schwartz**
Program in Media Arts and Sciences
May 6, 2005

Certified by: **John Maeda**
Allen Professor of Media Arts and Science
Thesis Supervisor

Accepted by: **Andrew B. Lippman**
Chair, Departmental Committee on Graduate Studies
Program in Media Arts and Sciences

Process Makes Perfect

Marc Schwartz

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on May 6, 2005, in
partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

This thesis presents a software architecture for developing art and design tools that automatically capture the artist's interaction and records the entire creative process. The software uses the artist's interaction with the digital medium as its true data-type and streams the data to a database. Over time, a large, searchable, knowledge bank of artisan technique would accrue. Querying this knowledge bank allows quick access to a wide array of user-interface interactions, artistic gestures, and annotations. Additionally, the design tools utilize annotation, search, and playback software to make viewing artistic processes richer and more useful. Using interaction with the medium as a searchable data-type and having useful means to view the data allows artists to query and review each other's artistic technique. This thesis applies this principle toward prototyping a new creative learning and tutorial software platform on the web.

Process Makes Perfect

Marc Schwartz

The following person served as a reader for this thesis:

Walter Bender
Executive Director, MIT Media Laboratory
Senior Research Scientist,
Program in Media Arts and Sciences

Process Makes Perfect

Marc Schwartz

The following person served as a reader for this thesis:

Mitchel Resnick
Associate Professor of Learning Research
MIT Media Arts and Sciences

Acknowledgments

I would like to thank my loving family. Mom, thank you for teaching me to be patient, care for people, think deeply about things, and be strong in the face of hardship. Dad, thank you for giving me the courage and support to chase my dreams, and the will to never compromise them. Sonya, thank you for your non-judgmental ear, your unwavering confidence in me, and all of the laughs over the years.

Sincerest thanks to Burak Arikan, James Dai, Ben Fry, James Seo, Seung-Hun Jeon, Atsushi Kunimatsu, Carlos Rocha, Noah Shanti Fields, Kelly Norton, Allen Rabinovich, Patrick Menard, Heather Pierce, Vinnie Russo, Hilary Karls and everyone at the Physical Language Workshop. I feel honored to have worked (and sometimes lived) in the lab with you. Special thanks to Sijia Mimi Liu. Shout out to my old friends from the Meadow and Loomis. Much love to my roommates, Ian and Lisa, for their friendship and understanding. Additionally, I am deeply indebted to the Media Lab for having met Andrea Chew there. Gnat, you fascinate and excite. So much fun and more to come!

Warm thanks to Walter and Marilyn Rabetz for lighting the fire and to Ettiene Delarciox who helped fuel it for a while. My deepest appreciation to my thesis readers Walter Bender and Mitch Resnick. The opportunity to work with the two of you on this thesis has made my MIT experience complete.

Much appreciation to all of my wonderful sponsors and The Simplicity Consortium. Special thanks to Toshiba and Samsung Corporations for making this work possible through their generous hardware contributions.

This Thesis was conceived in its final resting place, The MIT Rotch Library.

This thesis is dedicated to my mentor, **John Maeda**. My most he**art**felt gratitude for everything you have done **for** all of us at PLW. It will be **m**any year**s** before I fully di**gest** all that you have ta**u**ght and showed me. My expe**r**ience with you will no doubt b**e** the root of anything I manage to achieve in the future. Thank you for alway**s** dreaming big.

Table of Contents

Preface

Title
Abstract
Acknowledgments
Table of Contents
Table of Figures

1. Introduction

1.1 Motivation
1.2 Overview
1.3 Contributions

2. **Background**

2.1 Learning Processes
2.1.1 Brain Processing
2.1.1.2 Sensory modalities
2.1.1.3 Physical Needs
2.1.1.4 Environmental Preferences
2.1.1.5 Social Groupings
2.1.1.6 Attitudes
2.1.2 Multiple Intelligence
2.1.2.1 Visual
2.1.2.2 Verbal
2.1.2.3 Logical
2.1.2.4 Kinesthetic
2.1.2.5 Musical
2.1.2.6 Interpersonal
2.1.2.7 Intrapersonal
2.1.2.8 Naturalist
2.1.2.9 Diagnostics
2.2 Creative Learning
2.2.1 The Mind of the Artist
2.2.2 The Eye of the Artist
2.2.3 The Hand of the Artist
2.3 Learning Software
2.3.1 Quick References
2.3.2 Videos
2.3.3 Recipes
2.3.4 Screen Takeover
2.3.5 Wizards
2.3.6 An Electronic Book Prototype
2.3.7 Forums
2.3.8 Wikis
2.3.9 Wink
2.4 Foundation Experiments
2.4.1 Platform
2.4.2 Streamy
2.4.3 MIT Treehouse Studio Prototype

2.4.4 Comm-piler
2.4.5 Streamy Messaging Extension
2.4.6 Draw Together
2.4.7 Code Together
2.4.8 Record and Takeover
2.4.9 SMPL
2.4.10 Toward Treehouse 1.0

3. Treehouse Historian
3.1 Example Scenario
3.2 Implementation Details
3.2.1 Gesture Recording
3.2.2 The Historian
3.2.3 Historian Event Type
3.2.4 Process Annotation
3.2.5 Process Search and Retrieval

4. Evaluation and Analysis
4.1 What Can We Learn?
4.1.1 Artist 1
4.1.2 Artist 2
4.1.3 Artist 3
4.1.4 Conclusions
4.2 Further Analysis of Annotation Process
4.2.1 Data Summary
4.2.2 Conclusions
4.3 Functional Comparisons
4.3.1 The Intelligences of Treehouse Historian
4.3.2 Recipes vs. Process Annotation System
4.3.3 Tutorial Video vs. Process Playback
4.4 Feedback
4.4.1 Treehouse Paint Tool
4.4.2 Process Annotation
4.4.3 Playback
4.4.4 Search

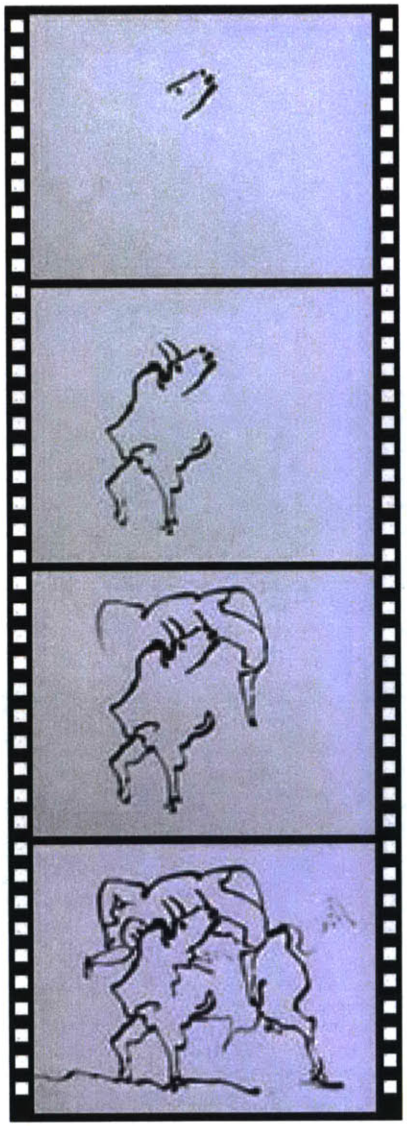
5. Conclusion
5.1 Future Directions
5.1.1 The Danger Mouse Scenario
5.1.2 Focus Group Scenario
5.1.3 UI Augmentation Scenario
5.1.4 More Active Interfaces
5.2 The Ninth Intelligence

Bibliography

Table of Figures

1. Frames from *Le Mystère Picasso*.
2. Two tools for artists.
3. The Author's multiple intelligences.
4. Drawing copied inside FMRI machine.
5. FMRI Images of Humphrey Ocean's brain.
6. FMRI images of a non-artist's brain.
7. Humphrey Ocean wearing eye tracker device.
8. Humphrey Ocean's eye fixations on four models.
9. Humphrey Ocean's hand movements.
10. Humphrey Ocean's pencil path.
11. Home made *Adobe Illustrator* Quick Reference.
12. Video from *www.learning-maya.com*.
13. A Lockstep tutorial recipe.
14. *Famatech Radmin*.
15. *Apple Balloon Help*
16. David S. Backer's E-book prototype.
17. A tutorial forum posting.
18. The *wikipedia.com* tutorial is a wiki itself.
19. Editing the *wikipedia.com* tutorial wiki.
20. *Wink* in Action.
21. Streamy is a simple web services architecture.
22. The TCP/IP RFID Reader.
23. Treehouse Studio user home page.
24. Treehouse Studio Photo Applet.
25. A Remote compilation architecture.
26. Comm-piler experiment.
27. The Draw Together System .
28. The Code Together Hierarchy Map.
29. The Code Together Editor.
30. A digital brush path.
31. Record and Takeover
32. The new Treehouse Studio Paint Program.
33. Creating key frames with Treehouse Historian.
34. Treehouse Historian Annotation Panel.
35. A Historian session at a glance.
36. Historian playback
37. Historian search feature.
38. Historian gesture capture architecture.
39. Historian database structure.
40. Annotations and thumbnails also link by time stamp.
41. The annotated key frames made by Subject 1.
42. A still frame captured during process playback.
43. Artists motor activity is immediately knowable.
44. Artist 1 draws the mouth with 2 lines.
45. Artist 1's completed drawing
46. Artist 2's key frames
47. Artist 2's process
48. The Artist used layers.

49. Artist 2's final painting of the still life.
50. Artist 3's key frames.
51. The Layer Panel
52. Artist 3's final drawing.
53. Annotation style comparison.
54. Number of annotations.
55. Length of annotations.
56. Percentage of time spent annotating.
57. Multiple Intelligences of Treehouse Historian
58. The Grey Album.



1. Introduction

1.1 Motivation

In 1956, Henri-Georges Clouzot had the idea to use film technology to capture the painterly process with Le Mystère Picasso. In one sequence of the film, Picasso paints the death of a matador. He first uses his brush to confidently define the contours of the forms. Then, working at a furious pace, he applies colors in layers of ink. After watching the sequence a few times, one can observe Picasso's aesthetic sense and clearly see the artist making design decisions and then witness the tangible result of each decision.

To me, this visual experience provided a different kind of education than art history lectures. I had seen a new art educational medium that was a work of art in and of itself. It inspired me to create works of my own and compare my process to the great master. Now that art can also be created in the digital domain, I wondered how a digital approach could extend Clouzot's idea. *Le Mystère Picasso* required special equipment and was a one-off experiment, I thought building Clouzot's idea into software could give us the opportunity of capturing the work of amateur and

professional artists all over the world.

All creative work can be described as an accumulation of interactions with a medium. In traditional arts and crafts, the artifact itself is often the only record of the process. Art historians in this century have used techniques such as x-raying and conventional detective work in an attempt to reveal the processes that created a painting. The master's brush can be examined, but there is often no method for studying exactly how it was used to create a beautiful piece of art. It can be difficult to figure out how many people actually worked on a piece. What tasks did the master perform? What tasks did his apprentices perform? How long did the piece take to complete? What were the key steps involved? These questions remain unanswered for many of the world's artistic creations.

The processes employed by artisans of the past must be reassembled through rigorous detective work, but digital art and design tools provide an unprecedented opportunity to capture an artist's process. All digital tools precisely transform physical gestures from mice, tablets, or other input devices into discrete numerical values. Numerical data sampled from the input devices are streamed and instantly consumed by the art application.

Adobe Photoshop has a feature called history where by this data is stored for the purpose of undoing actions, but the data is not stored upon saving the file. Storing this history data in a database creates a fundamental learning opportunity because a new user could be informed of another artist's process. Implementing tools to help query and view these processes would help artists learn valuable skills from each other.

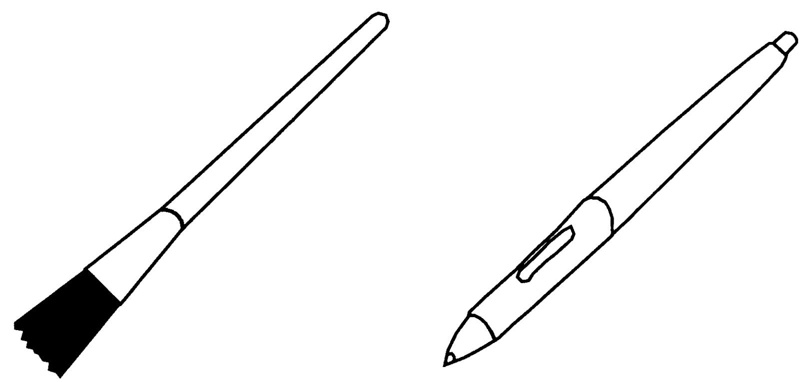


Figure 2. Two tools for artists. The second allows the capturing of its use history.

1.2 Overview

The remainder of this thesis is organized into four chapters and a bibliography.

Chapter 2, *Learning Processes*, surveys different learning styles and explores the cognitive principles of creative learning. Chapter 2 also includes a broad summary of learning software technologies presently in use.

Chapter 3, *Treehouse Historian*, describes the usage and underlying technology of a reference implementation.

Chapter 4, *Evaluation and Analysis*, attempts to qualitatively and quantitatively evaluate the reference implementation.

Chapter 5, *Conclusion*, contains possible future directions of this work and final thoughts.

1.3 Contributions

The goal of this thesis is to develop a new learning and tutorial platform for digital design software and present historic, scientific and analytic context for evaluating it's relevance and usefulness. In realizing this goal, this thesis will contribute:

1. Survey of cognitive learning patterns and as they apply to acquiring creative skills.

2 Summary of current software tutorial technologies.

4. Technical descriptions of crucial background software development performed during the course of my study.

5. Description of Treehouse Historian. The reference implementation of this thesis.

6. An analysis and evaluation of the Historian system.

7. Conclusions and directions for future experimentation.

2. Background

The following chapter briefly summarizes background research on the process of learning. This investigation aims to document exploration of learning theory. Section 2.1 will outline some popular learning theories. Section 2.2 will discuss several experiments addressing the topic of creative learning. Section 2.3 will review several learning technologies that were influential to this thesis.

2.1 Learning Processes

The findings in this section are summarized from secondary research sources. The information provides context and offers some criteria for evaluation.

2.1.1 Brain Processing

Brain processing describes the way in which the brain is able to take in and perceive information. Typically, a learner possesses a bias toward either left brain or right brain processing. The learner's dominant half-brain has considerable impact on their learning style as different learning strategies must be applied to maximize their potential.

The left-brain is known as a successive processor. This means that left-brain dominant learners acquire knowledge step by step. These types of learners prefer starting out with details and gradually lead up to a total understanding of a particular skill. Left-brain dominant learners are also referred to as analytic. They research topics on their own, respond to logic, plan ahead, speak with few gestures, learn better from lectures, prefer bright lights while studying, and concentrate mainly on what the experts have to say [Freedman 2001]. Analytic learners take pride in obtaining facts so that they may understand certain concepts and processes more easily [McCarthy 1987].

The right-brain, on the other hand, is known as a simultaneous processor. This means that students who are dominant in their right-brain learn the general concept first and then go back to learn the specifics of that topic. Right-brain dominant learners are also known as holistic or global learners. These learners respond to emotion, recall people's faces, use gestures while speaking, prefer some type of sound or music while studying, and take in information in a varied order [Freedman 2001]. Global learners are the type of students who retain large amounts of information and gain understanding through groupings and connection making. Holistic learners really understand and recognize how the mind, body and spirit are all connected [McCarthy 1987]. Holistic learners view learning as an experimental process. They often have success taking a bird's eye view approach and zooming in on only the specific details that interest them.

2.1.1.2 Sensory modalities

Sensory modalities include auditory, visual, tactile, and kinesthetic learning styles. An auditory learner, understands information through talking or hearing about it. An auditory learner considers the tone of voice, pitch, and speed. These learners enjoy class discussions, lectures, and listening to how others interpret the topic [Prashnig 1998]. Auditory learners benefit greatly through reading information out loud or by using a tape recorder.

A visual learner, on the other hand, gains knowledge about a particular topic by either reading, seeing, or visualizing the information at hand. Visual learners need to see body language and facial expression in order to understand the material being taught. These learners benefit from sitting in the front of the class avoiding visual obstructions from the teacher. They learn better from visualizations such as diagrams, videos, and other displays [Paulson 2001]. Visual learners tend to make extensive use of their notebooks. In their most attentive state, they make diagrams an sketch out the material. In less attentive learning states they tend to doodle.

A tactile or kinesthetic learner acquires information by touching, feeling and doing. These types of learners respond very well to experiential or the hands-on learning approach. Kinesthetic learners deeply enjoy the process of exploring everything around them [McCarthy 1987]. Conversely, these learners have a hard time sitting still and become easily distracted when they are not directly involved in an activity.

2.1.1.3 Physical Needs

A student's physical preferences include mobility, intake, and time of day. Mobility has to do with one's own preference to get up and walk around or remain stationary while learning about a particular topic. Intake, on the other hand, has to do with whether or not the student needs to eat, drink, snack, or chew while studying. Finally, the time of day is another important factor that affects how a student learns best. For some students, their *peak time*, simply meaning the time in which they are able to concentrate the best, may be at 1:00 o'clock a.m., while for others it may be 7:00 o'clock at night. [Prashnig 1998]. It is important for the student to discover their peak time so that they will be able to maximize their intake during learning or studying.

2.1.1.4 Environmental Preferences

This area has to do with a student's own preferences of the surrounding sound, light, temperature, and work area. It is important for some students, most likely global and holistic learners to always have music or sound of any type going on in the back round for them to most effectively learn. On the other hand, analytic learners prefer to work where it is quiet and there are no distractions, in order to achieve their greatest potential. As for light, again, some learners prefer bright lights while studying, while other may insist on dim lights or even candlelight. Temperature is also an important part of learning because for some, a warm environment is preferable for acquiring information, while for others, a cold environment enables a student to stay awake and therefore has an effect on better learning. Finally, work area also has a lot to do with the way one acquires information [Prashnig 1998]. Analytic learners prefer formal design when learning over informal, which simply means a desk and a chair as opposed to a bed or a couch. Global and holistic learners prefer comfort. A hard desk and a chair don't allow them to concentrate on what they are learning.

2.1.1.5 Social Groupings

Social groupings include the individual's preference to work alone, in a pair, with friends, or in a team. Depending on what the student is more comfortable with, his or her work may greatly benefit from that preference. However, many students also prefer to learn directly from an authority figure, meaning from either a parent or teacher, and that is also a good choice. Either one of these options, if available to the individual, would be beneficial and enhance the learning experience.

2.1.1.6 Attitudes

The sixth and last factor that has a deep impact on a student's learning performance is in fact the individual's attitude, including motivation, persistence, conformity, structure, and variety. When a student is motivated to learn, it is easier for them because he or she have made the decision to do whatever is required to understand what is being taught. Some students have a strong persistence that allows them to pay great attention and complete their task at hand [Prashnig 1998]. Conformity is also an important part in a student's learning process because if the student is more of a rebellious nature, they will have a more difficult time focusing, and therefore a more difficult time learning. Structure, on the other hand, has to do with whether or not the student is able to learn on his or her own, or if guidance from others is necessary for him or her to achieve their goals. Finally, variety is also a key aspect in a student's learning experience [Prashnig 1998]. Some individual's need changes or variety, while other prefer sticking to the same routine over and over again because they feel as though that is the best way for them to learn.

2.1.2 Multiple Intelligence

Dr. Howard Gardner, professor at the Harvard Graduate School of Education introduced the Theory of Multiple Intelligence. The theory divides intelligence into eight discrete categories. These categories present a general framework, but each individual's mind combines these intelligences in it's own unique way. Figure 3 shows the results of a diagnostic test taken by the author of this thesis. Certain types of intelligence are dominant in this individual, but all are essential. The theory has become the model for many recent educational strategies and has affected the way education is evaluated.

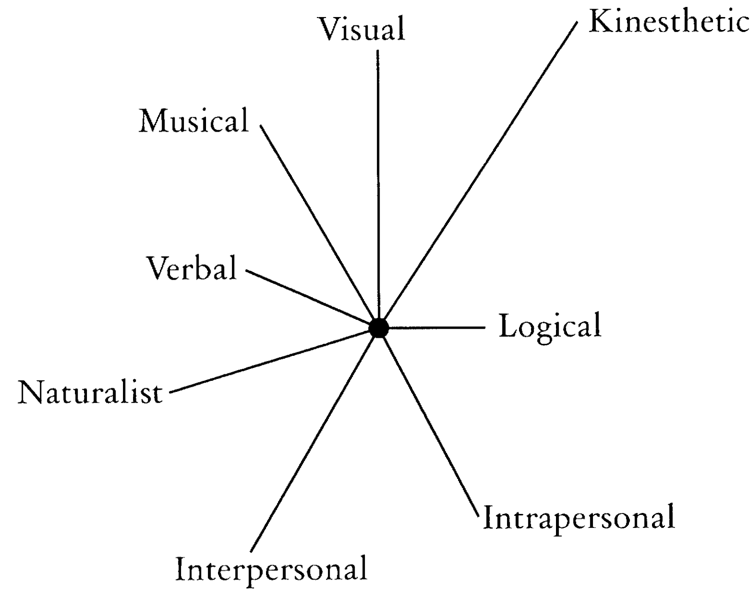


Figure 3. The author's results for a multiple intelligence diagnostic test. Longer the axis, the higher the score. My most dominant intelligence is kinesthetic, my least is logical.

2.1.2.1 Visual

"Visual intelligence" refers to the ability to recognize things in the visual or spatial domain. In order for these types of learners to retain information, they need to think in pictures and generate mental images. These individuals enjoy movies, maps, charts, and pictures. They excel in activities involving spatial relationships, including: sketching, puzzle building. As for their careers, it would be beneficial for them to consider becoming sculptors, inventors, architects, mechanics, or engineers [plsweb 2004].

2.1.2.2 Verbal

"Verbal intelligence," which is merely the ability to understand and use words and languages. These types of learners are usually good speakers. Unlike visual intelligence, these learners think in words instead of pictures. They enjoy listening, writing, storytelling, and teaching. At times they are very humorous, and have the ability to convince others of their point of view [Gardner 1993]. Students who consider themselves verbal learners should think about becoming poets, journalists, teachers, lawyers, or translators [plsweb 2004].

2.1.2.3 Logical

"Logical intelligence" is the third of these eight Multiple Intelligences and individuals who fall into this category have an immense ability to use reason, logic, and numbers. These types of learners are curious about the world around them, just like tactile and kinesthetic learners are. These students also ask lots of questions and just like abstract active learners, enjoy engaging in different challenges and experiments. They dominate in problem solving. [Gardner, Hatch 1989]. Doing controlled experiments, performing difficult mathematical calculations, working with geometric shapes, and organizing information. Students with logical intelligence might consider becoming scientists, researchers, accountants, or computer programmers [plsweb 2004].

2.1.2.4 Kinesthetic

The fourth of these Multiple Intelligences is known as "kinesthetic intelligence." Individuals who fall into this category are able to handle objects with great skill and control body movements. They also have a good sense of balance and hand-eye coordination, and seem to express themselves through movements [Gardner 1993]. These types of individuals shine in dancing, sports, crafts, acting, and building. Students with kinesthetic intelligence might consider becoming athletes, phys-ed teachers, dancers, actors, or firefighters [plsweb 2004].

2.1.2.5 Musical

"Musical intelligence" is the fifth type of Multiple Intelligence and students who associate with this type of intelligence have a great ability to produce and appreciate the beauty of music. These learners think in sounds, rhythms, and patterns. They are either very critical of what they hear or just the opposite, very pleased. These individuals are also very aware of the sounds going on around them, not only having to do with music, but in nature as well. Many have an understanding of the structure and rhythm of music, and can point out tonal patterns. Students with musical intelligence might concentrate on becoming musicians, disc jockeys, singers, or composers [plsweb 2004].

2.1.2.6 Interpersonal

The sixth of these Multiple Intelligences is called "interpersonal intelligence." Students who fall into this category easily relate and understand others. They have the ability to sense feelings, intentions, and motivations. They enjoy organizing and try and see things from other's point of view in order to understand what they are actually feeling. Sometimes these individuals can be very manipulative but at the same time try to promote peace and encourage cooperation [Gardner 1993]. They use both verbal and non-verbal language to communicate with others. They are very good at listening, using empathy, noticing mood changes in others, building trust and forming positive relationships with others. Students who relate to this type of intelligence, might consider becoming counselors, salespersons, politicians, businesspeople [plsweb 2004].

2.1.2.7 Intrapersonal

"Intrapersonal intelligence" is the penultimate type of Multiple Intelligences discussed here. Individuals who fall into this category have the ability of really understanding their true self and their inner state of being. They concentrate on dreams, relationships, and their own personal strengths and weaknesses. They are very good at reasoning with themselves, analyzing themselves, evaluating their own thinking patterns and understanding their role in the relationships they are in [Gardner, Hatch 1989]. Students who can relate to this type of intelligence make good researchers, theorists, and philosophers [plsweb 2004].

2.1.2.8 Naturalist

Finally, the eighth and final type of Multiple Intelligence is known as "naturalist intelligence," and students who associate with this type of intelligence have the ability to distinguish and categorize flowers and plants. [Gardner 1993]. They are fascinated with animals and their behavior. These types of people might consider becoming farmers, or biologists [plsweb 2004].

2.1.2.9 Diagnostics

After learning about the different learning styles, patterns, and the theory of Multiple Intelligence, it becomes apparent that each student operates in their own personal manor. Each individual is made up of a very complex combination of these intelligences. Discovering the type of learner one is, allows them to understand themselves better and make the best out of their learning experience. A student is then able to alter their learning approach to maximize their study habits [Manner 2001]. Multiple Intelligence is not a theory designed to quantify and difinitively catagorize students, rather it serves as a tool to understand and engage each student's individuality.

2.2 Creative Learning

Betty Edward's drawing tutorial book, *Drawing on the Right Side of the Brain*, incorporates brain research into the formulation of the teaching methods. Drawing is considered a right-brain activity as it involves working from the whole. In addition, developing a drawing, choosing the elements, matching and mixing colors, placing in the shadows and highlights requires strong simultaneous processing skills as one must work on various things simultaneously. These activities place the artist in a highly visual and kinetic sensual state. The right-brain is highly dominant in the physical act of applying the pencil to paper, but the left-brain is also crucial to the creative process. The left brain enables the artist to look critically at their work and to be analytical. Art schools recognize the necessity of developing both hemispheres as much of studio class time is devoted to critique.

2.2.1 The Mind of the Artist

From 1991 to present, Scientist and Filmmaker, John Tchalenko, has been studying Painter, Humphrey Ocean. In collaboration with Chris Miall, they are using advanced biomedical tools to explore how an artist paints a portrait. Tchalenko describes the picture production process as "hundreds of eye-brain-hand-eye cycles, each lasting a few seconds or less" [Tchalenko 2001].

While lying inside an FMRI scanner, Ocean performed the following experiment. A non-artist was asked to complete the same procedure. Given a small notebook containing two portrait photographs and two abstract designs, each made a thirty second study of the four pictures. By comparing the brain activity when copying the designs to that when drawing portraits, the processes exclusive to drawing a face could be seen.

Figures 5 and 6 show the brain activity by Ocean and a non-artist during the drawing of the four images shown in Figure 4. The image shows increased blood flow measured along near horizontal slices. The FMRI scan displayed were mapped onto an image of the brain.

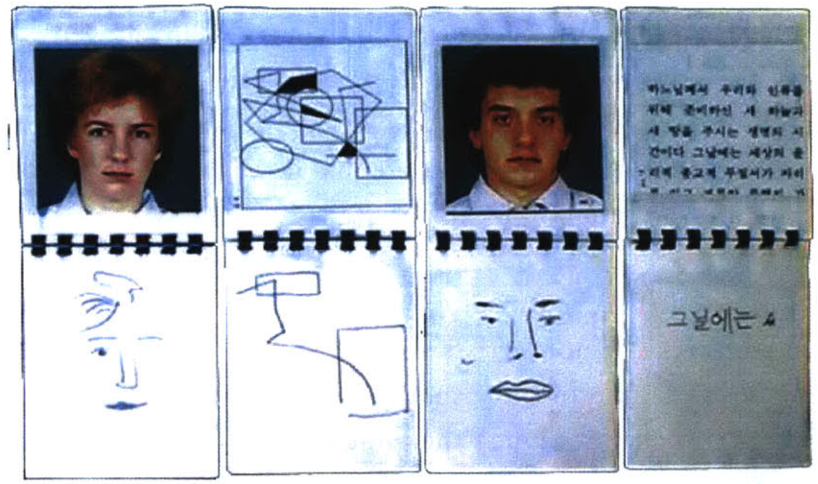


Figure 4. [Tchalenko, Miall 1998] The drawings copied.

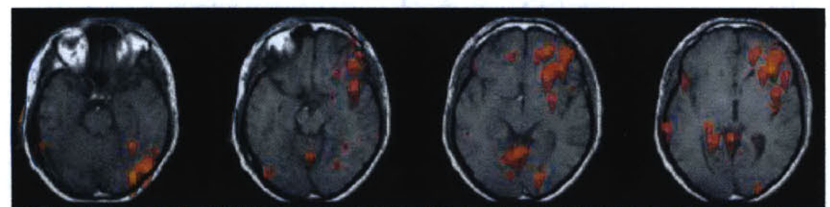


Figure 5. [Tchalenko, Miall 1998] FMRI scans of Ocean's brain.

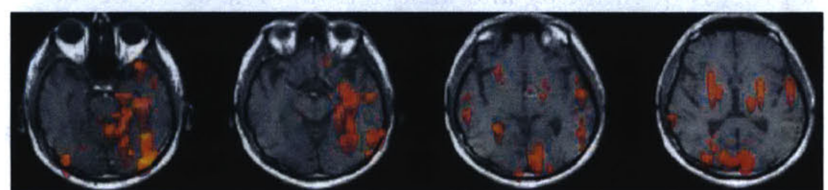


Figure 6. [Tchalenko, Miall 1998] FMRI images of a non-artist's brain.

Humphrey Ocean (Figure 5)
With the exception of the first drawing, Ocean showed no activation in the visual cortex, but in more frontal regions, suggesting that he was relying on an abstracted representation of each photograph. He was "thinking" the portraits [Tchalenko, Miall 1998].

Non-artist (Figure 6)
In contrast, the non-artists showed most activation in the posterior region of the visual cortex, indicating that they were "slavishly copying" the photograph [Tchalenko, Miall 1998].

The comparison of Ocean and a non-artist for the same task revealed that although entirely right-brained, abstraction of form does not involve the visual cortex. This process of abstraction (in this case, simplification) is used by a trained artist while copying. This suggests that even when working representationally, art is largely a process of thinking and abstraction. The artist bends forms, breaks up shapes, and give objects unlikely textures or colors. Artists make these transformations in an effort to communicate. This is a much less objective process than that of the non-artist. The non-artist aims to mimic a copying machine by short-circuiting the eye and hand. The artist relies on hand-eye coordination only after thinking through the abstraction.

2.2.2 The Eye of the Artist

Tchalenko and Miall conducted extensive interviews with Ocean, who was able to describe his process with specific details.

"I will start with what I can see from where I am. I try to achieve a likeness. But what I want is a likeness to the reaction I have to something I can see... I can't separate what [the subject's] hands are like, from what they look like, from what I feel about them." [Ocean 1998].

Figure 7. [Tchalenko, Miall 1998] Humphrey Ocean at his easel wearing Miall's eye tracker device.

Here, Ocean alludes to a phenomena that many artists encounter, synesthesia. The experience

Figure 8. [Tchalenko, Miall 1998] Humphrey Ocean's eye fixations on four models.

perceived by one sense (vision in this case) can trigger other sensual experiences. Ocean further describes his experience:

"I'm sure of what I am seeing, I'm not quite sure what I'm going to do about it. So I make a decision. The final result is made up of a great many decisions." [Ocean 1998].

Tchalenko and Miall studied Ocean's artistic decisions in depth. Using an eye tracker proved to be an important technological technique.

Ocean first chose his model out of four candidates (Figure 8) whom he was seeing for the first time. His eye explored each face with a very rapid series of short fixations, at a rate of 140 fixations per minute, each fixation lasting an average of 0.4 secsonds.

"I probably do tend to start with the eyes. I would say that that's no different when I'm drawing to when I am talking to somebody. When you're bicycling, the way to get the attention of a driver in a car - you make eye contact, and then they will watch out for you. It's not an art thing... it's about survival." [Ocean 1998].

Ocean describes eye contact as a means of speechless communication. If you think of eye-to-eye fixations as words, it is easy to tell which model he communicated with the most.

It may come as no surprise that Ocean selected the fourth candidate, Nick, and drew his portrait on the following day. The eye-tracker was used again during the drawing process.

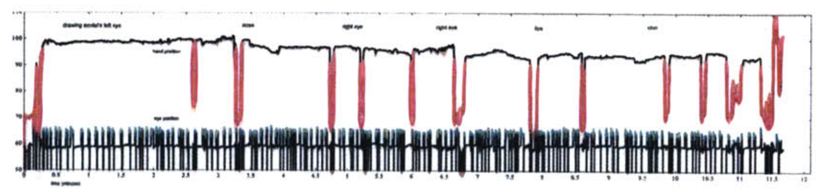


Figure 9. [Tchalenko, Miall 1998] This graph maps Ocean's hand movements. The large dips describe moments where hs hand rests at his lap.

The eye movement patterns may vary, but the rhythm remains fundamentally unchanged with about 12 fixations on the model each minute. Fixations were of the order of 1 second. Long fixations are quite different from normal eye movements. When Humphrey is not drawing, his fixations are briefer and more frequent: 0.4 seconds and 140 per minute.

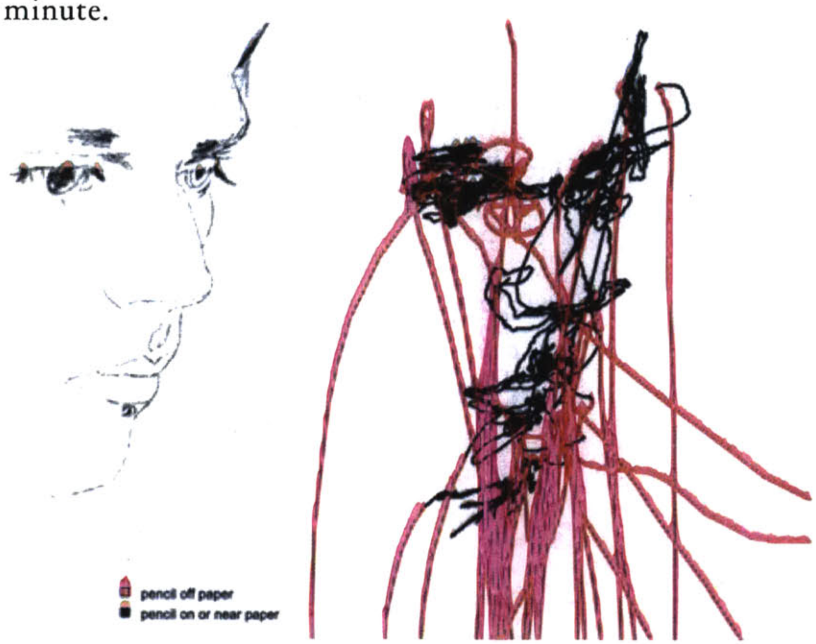


Figure 10. [Tchalenko, Miall 1998] Humphrey Ocean's pencil path during study of Nick. The paths indicate moments when the pencil is on or near the paper, as well as moments where the pencil is off of the paper. Part of Ocean's process for precision is practicing drawing a line without making marks.

2.2.3 The Hand of the Artist

During the 12 minute drawing of Luke 2, Humphrey fixated the model about 150 times and paused his hand near his lap 11 times.

In this experiment, clues about the purpose of the extra hand movements were found when carefully examining the data for the drawing of Nick's lips in the 5 hour portrait.

Humphrey often practiced each line, with the pencil just off the paper, before drawing it. These repetitive movements of the eye and the hand, each apparently informing the other, often lasted for 10 or 20 cycles.

In all my work I'm after precision, and I think detail is precision... Detail means where the line lands, and if it lands a millimeter to the right or a millimeter to the left, it changes the weight, in some way, of the shape that it is describing. So when that line lands, you just want it to land in the right position, whatever that is. [Ocean 1998].

Mostly the eye followed closely the hand's movements, but the exceptions were of interest: systematically, the eye fixated the model as the hand approached the paper after a pause, and occasionally, while the hand was practicing, the eye glanced back at the model, presumably when the painter needed to refresh his memory [Tchalenko, Miall 1998].

2.3 Learning Software

The previous section summarized literature reviewed in perpetration for this thesis. The intent was to approach software writing with influences outside the field of Computer Science. The remainder of this thesis, however, aims to discuss technology. This section will discuss current software packages that facilitate learning scenarios.

2.3.1 Quick References

"Quick References" aim to summarize the various functionalities of a software system. These are often designed for persons with at least intermediate knowledge. In many cases, users make their own quick references so they may quickly recall commands and tool functionalities. (Figure 10), an annotated *Adobe Illustrator*

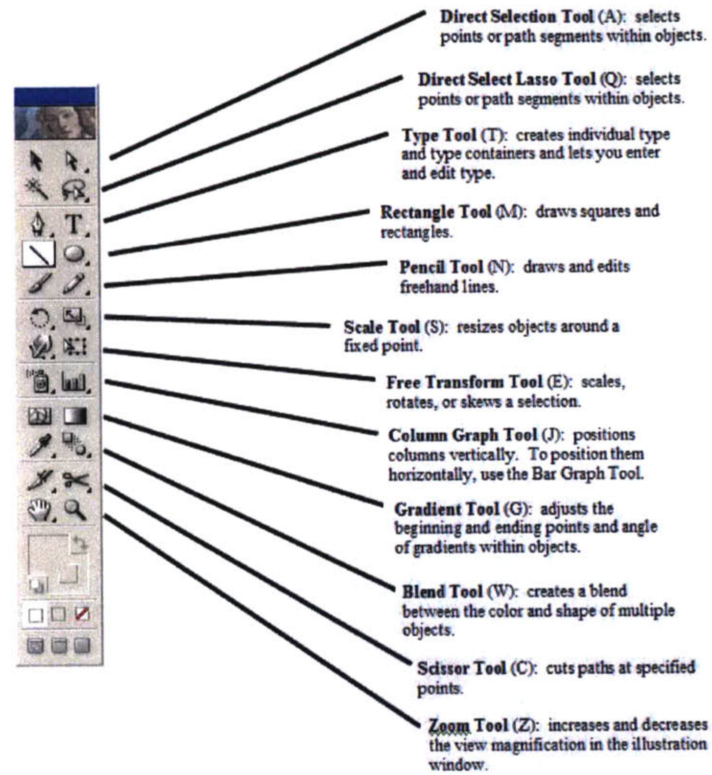


Figure 11. An Adobe Illustrator Quick Reference created by a night-school art teacher.

toolbox reference was created by a continuing education teacher at the Rhode Island School of Design and could readily be found near workstation mouse-pads. Quick References help in recalling software functions, but supply little help in learning artistic technique.

2.3.2 Videos

Annotated video screen captures (Figure 8) are recordings of teacher's real-time interaction with creative tools. These captures are produced as digital movies and usually include a voice track where the teacher can annotate their process. These films present the process in a lockstep fashion from beginning to end, but give the student the ability to slow down, speed up, and seek. The company, *Quick Click Training* produces a very popular series of educational videos for all types of creative software. These films consist of a teacher talking through a lesson. The image consists only of the teacher's screen. Students usually interact with these videos in a linear manner by replicating the teacher's steps. Students can observe both the operation of the software and the artistic skill of the teacher.

2.3.3 Recipes

Recipes are text-oriented lock-step instructional materials. In general, the location of these documents have moved from books to locally stored digital "readme files," to web based media. A typical web based tutorial like the Photo-

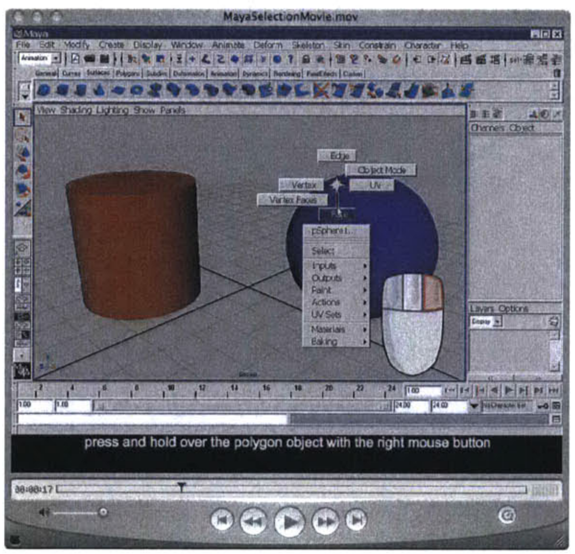


Figure 12. This video from *www.learning-maya.com* includes an overlay showing key mouse button presses. This is particularly important for systems with mouse intensive user interfaces using multiple mouse buttons.

1. Make the red octagon. In order to do this, you have to tell PS exactly what you want: a red, 8-sided polygon which has the path with it. (This is called a shape layer.)
Choose your shape tool and then choose Polygon from the options.
In the options bar, just to the right of your geometry options, there is an arrow. Click that and make sure that there are no boxes checked. Click the arrow again to turn off that box.
Where it says "sides," put an 8 in there.
Click the Shape Layer button to the left in the Options bar.
Choose red for your foreground color.
Now put your cursor on the canvas and drag out an octagon. Make it bigger than you think you need. As you drag, notice that you can angle it however you want. (If you hold the Shift key

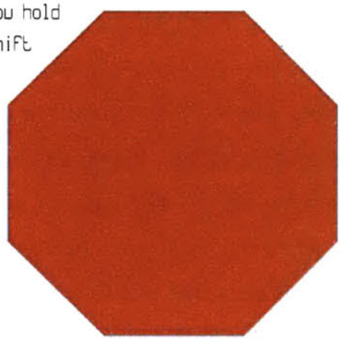
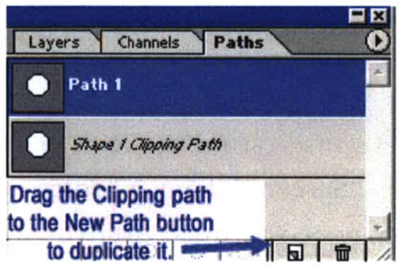
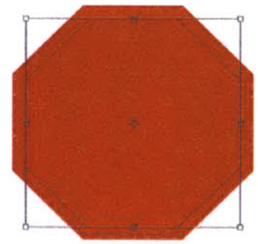


Figure 13. Lots of text! An example of a lockstep tutorial recipe from *www.myjanee.com.*

here, what happens to your orientation?) Before you release your mouse, line up the octagon so that the top bar is perfectly horizontal.
Ctrl-S to save.**2**. Make the white line.
Now we are going to take the path that is around the red octagon and use it to make the white line around the inner rim. We could make a new path, but by using the



old one and transforming it, we will make sure to get the exact angles. First we need to duplicate the Layer clipping path.
Click your red octagon layer so that it is active.
Now click the Paths palette and then drag the Shape 1 Clipping path (In v. 7, it is called Shape 1 Vector Mask.) to the New Path icon to duplicate it.
Choose the Shape tool (or any other path tool) in the toolbox. Now Ctrl-T will bring up your Free Transform Path box.
Now you are going to drag this path inward till it is where you want your white to be.
Hold the Shift and Alt keys whilst you are dragging (and hold them down till after you release your mouse). (Shift keeps the proportions intact and Alt makes the transformation stay



centered.) Click Enter to transform your path.
Note: Sometimes, inexplicably, the whole red octagon will get smaller, instead of just the path. If this happens, hit Esc and then go to Edit > Free Transform Path... and work from there.
Click on the Layers tab to bring back the Layers palette. Make a new layer by clicking on the New Layer icon in the Layers Palette.
Choose white for your foreground color.
Click your paintbrush and choose the width of hard-edged (dot) brush that you want your white line to be. (In v. 7, click the brushes palette to be sure that you have no brush dynamics set.)
Now, see that little button at the bottom of the Paths Palette that is colored in.. and then there is one next to it that is NOT colored in. Push the one that is not colored in. That is the "Stroke Path" button. It will go around your path with whatever brush you last picked in whatever color onto whatever layer you had selected in your layers palette!
Click in your paths palette below your path in an empty space to make your path disapear.
Ctrl-S to save.
Note: If you last used some other tool than a brush, your stroke will be a nasty jaggy black line. If this happens, Ctrl-Z to undo and then choose a brush and do again.
Another thing: If you forgot to make yourself a new layer or if you don't have the layer selected in your layers palette, it can't stroke the path just into the air, so the stroke button will be grayed out.

3. Make the lettering.
Choose a better font than I did to do the Lettering.
And finally... Ctrl-S to save. (Yes, I know that i'm fanatical about saving. As you get more into this, you will be happy if you are in the habit of saving early and often. I hope to help instill this habit in you. :))

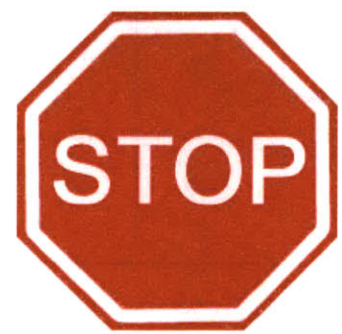


Figure 14. *Famatech Radmin.* An enterprise remote administration tool.

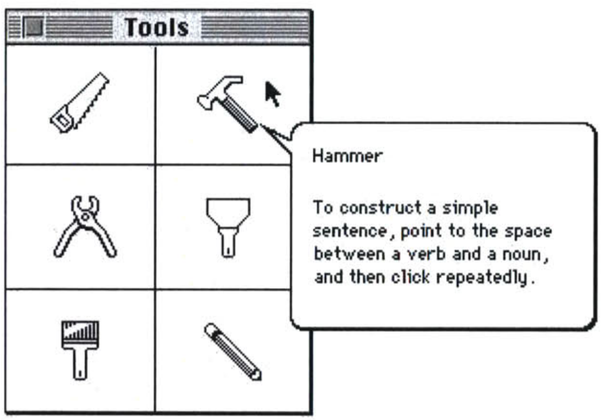


Figure 15. *Apple Baloon Help*

shop tutorials on *www.myjanee.com* (Figure 13) manifest as meticulous sequentially structured html pages. A learner would generally follow the instructions in a linear and lockstep fashion and try to repeat each step in succession. These tutorials aim to teach a specific process from beginning to end.

2.3.4 Screen Takeover

An interesting network oriented cousin to screen captures is "Screen Takeover" software This screen-interaction data may be streamed over the web instead of stored as a video file. Screen takeover software allows a system administrator to take control of a users computer in order to administer and configure it. The user whose computer is being configured can watch the result of the remote administrator's actions. These tools typically are not designed with education in mind, but if the user has a quick eye and strong memory, they may learn how to do the administration task themselves next time. *Microsoft Windows* and *Macintosh OS X* both implement a remote desktop; many standalone packages exist also including *Admin Magic*, *VNC*, Symantec's *PCAnywhere*, and seen in Figure 13, *Famatech's Radmin.*

2.3.5 Wizards

Software "Wizards" allow the user to do the configuring and administration. Wizards, like tutorial documents, offer step-by-step instructions in words and pictures. The key innovations that wizard offer is the ability to interact with the piece of software it is teaching the user about. For example, the *Microsoft Windows Add Printer Wizard* actuates menus and dialogs in the operating system itself. *Apple Balloon Help* is another example of this kind of integrated lockstep help system.

2.3.6 An Electronic Book Prototype

In 1988, David Backer presented a PhD dissertation on a new kind of electronic book for learning. Backer's invention, called The Movie Manual, is a training and teaching system that uses personal computing and optical videodisk storage to give its user access to text, sound, and images. The project combined audio, video and text into an extremely new an unique learning tool. The Dissertation introduces many of the technical and theoretical concepts of this thesis such as spectator participation in art [Backer 1988]. Backer's system discusses annotation as a viable future direction for his work.

"A natural impulse, and the first to be included as an option, should be the ability for the reader to "make notes in the margin"... the reader could make hand drawn notes and sketches, text highlights, reminders, comments, even jokes. The system would simply store the hand-drawn notes by recording the X,Y coordinates in a stream and associate them with the current page" [Backer 1988].

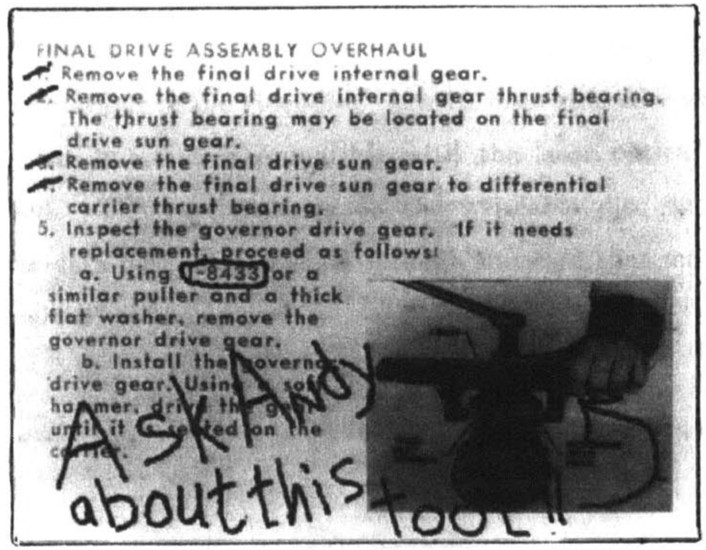


Figure 16. David S. Backer prototypes an annotation interface as part of the "Future Works" in his Electronic Movie Manual Dissertation. Backer suggested that the notes be handwriten. This is great for personalization and comfort, but makes searching more complex.

2.3.7 Forums

Lockstep tutorials involve the explication of a linear process, but other help software has a less linear structure. User Forums such as *www.photoshoptechniques.com* have a question and answer structure. The questions and answers tend to be specific in nature. Furthermore, the questioner and answerer may go back and forth a few times. In addition, any number of people may reply to a forum thread, so different perspectives are purveyed. The forums, usually archived and searchable, often provide a large specific knowledge base on an application. Forums exemplify a type of online help community because of the fact that users compose all of the content that's on the site. The more numerous and active the users are, the richer and more helpful the site becomes. Forums typically manifest as dynamically generated html pages. Since HTML serves as the target output format, the body text of each forums can be searched by web search engines such as *Google* and *Yahoo*.

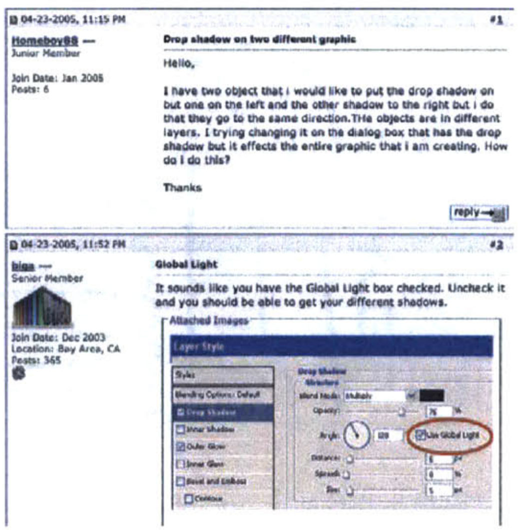


Figure 17. The *photoshoptechniques.com* forum gives users the ability to place images in their forum posts.

2.3.8 Wikis

Wikis [Leuf, Cunningham 2001] provide another searchable collaborative knowledge space. A wiki is a web site where every visitor has the power to become and editor. These differ from forums because the editing all takes place on one document. Forums manifest as a collection of question and answer fragments, while wikis reach a consensus thought the constant input of hundreds and sometimes thousands of editors. This style of editing may seem chaotic, but analysis of the popular wiki, *Wikipedia*, "revealed effective self-healing capabilities" [Viégas, Wattenberg and Dave 2004].

Figure 18. The *wikipedia.com tutorial is a wiki itself.*

Figure 19. Editing the wikipedia.com tutorial wiki.

Wikis particularly shine when pertaining to subjects that incrementally accumulate or accrue. Few tutorial wikis exist on the net, but one can find many wiki indecies that point to other types of tutorials. This model proves to be superior over a static HTML page linking to tutorials because users can easily add additional tutorials at any time. In addition, the tutorial poster as well as other users can post comments on the link wiki after reviewing the tutorial material. Mini reviews or summaries of the linked tutorial seem particularly useful here.

2.3.9 Wink

Wink [Kumar 2005] is a freely available and extremely powerful tool for creating software tutorials. Wink does not render screen captures into movie files, but instead allows tutorial makers to manually control the capture process. The user presses a special key each time they wish to capture a frame. Once the user has captured all desired frames, they are loaded into an annotation editor where text can be created and inserted. The cursor is separated from each screen capture so it may be repositioned or hidden. In addition, wink includes tools to make more compelling tutorials such as explanations boxes, buttons, and titles. Wink can output to a wide array of file formats for easy distribution on the Web. The simplicity of wink and the small file sized presentations aims to make tutorials flourish on the Internet.

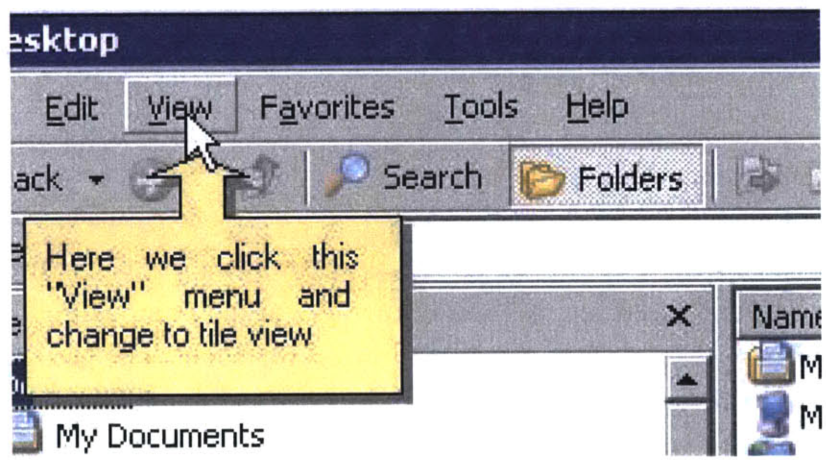


Figure 20. Wink in action: A n explanation box is used to add an annotationin a homemade *Microsoft Windows Explorer* tutorial.

2.4 Foundation Experiments

The folowing sections outline the technical building blocks that provide the foundation of this thesis. Without these supporting technologies, the thesis experiments could have never been completed.

2.4.1 Platform

All of the experiments detailed in this chapter make use of at least some Java technology. *Sun Microsystems* developed the Java platform in the early Nineties to provide a platform-independent software runtime. In adherence to their "write once run everywhere" philosophy, the Java programming language can be compiled down to object code instructions that can be executed on a platform-specific virtual machine.

In 2004, the Physical Language Workshop set out to create a TCP/IP based infrastructure for networked computing and chose Java as the implementation platform. The goal was to define and implement architectures to enable easier development of applications leveraging network based storage, cluster computation, and numerous communication architectures. Java was chosen for three main reasons:

- Platform independence
- Web Services support
- Server and Client Programming APIs

The third criteria became very important because students would have to only know or learn one language to start contributing. Physical Language Workshop (PLW), and other research groups within the MIT Media Lab, often employ many undergraduate researchers who join projects for a brief time only. Being able to learn quickly the prerequisite knowledge, begin development, and accelerates them towards a deep and rich learning experience.

2.4.2 Streamy

One of the earliest goals of PLW was to find infrastructure to support web delivered applications and TCP/IP based standalone devices. "Streamy" provided extensible architecture for creating simple web services. The request would be a URL string, and the response would be a data stream of appropriate MIME type. In this mode, the only requirement for devices was a functioning TCP/IP protocol stack. The second mode provided object transmissions specific to Java applications and applets. This model specifies that both inputs and outputs are serialized Java objects. In both cases, Streamy could connect to a centralized database in order to store files persistently, perform searches, retrieve data, and stream it to the appropriate client.

Most projects using Streamy made use of applet-to-server,

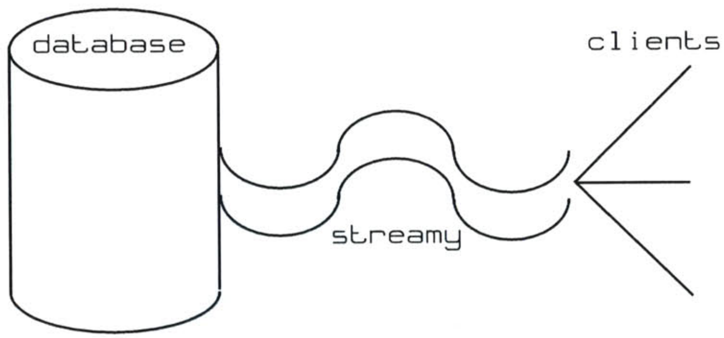


Figure 21. Streamy is a simple web services architecture.

server-to-applet, server-to-browser, and browser-to-server communication. In addition TCP/IP-based hardware such as the RFID reader [Holtzman, Rocha and Fields 2004].

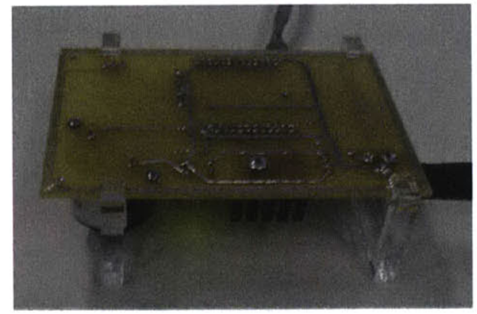


Figure 22. The *TCP/IP RFID reader [Holtzman, Rocha, Fields 2004] talks to a variety of web applications through Streamy.*

2.4.3 MIT Treehouse Studio Prototype

"The Treehouse Studio is a suite of online-accessible applications for drawing, painting, photography, sound-editing, and video-editing. By virtue of being situated on the Net, communities of content creation can be formed around any subset of the tools. Each software tool is written to be extensible with our back-end engine for the on-demand compilation of new features. Content created in Treehouse Studio is meant to be used primarily for our OpenAtelier project. Impact areas for the Treehouse Studio include distance education, online communities, digital entertainment, and creative knowledge management" [Maeda 2004].

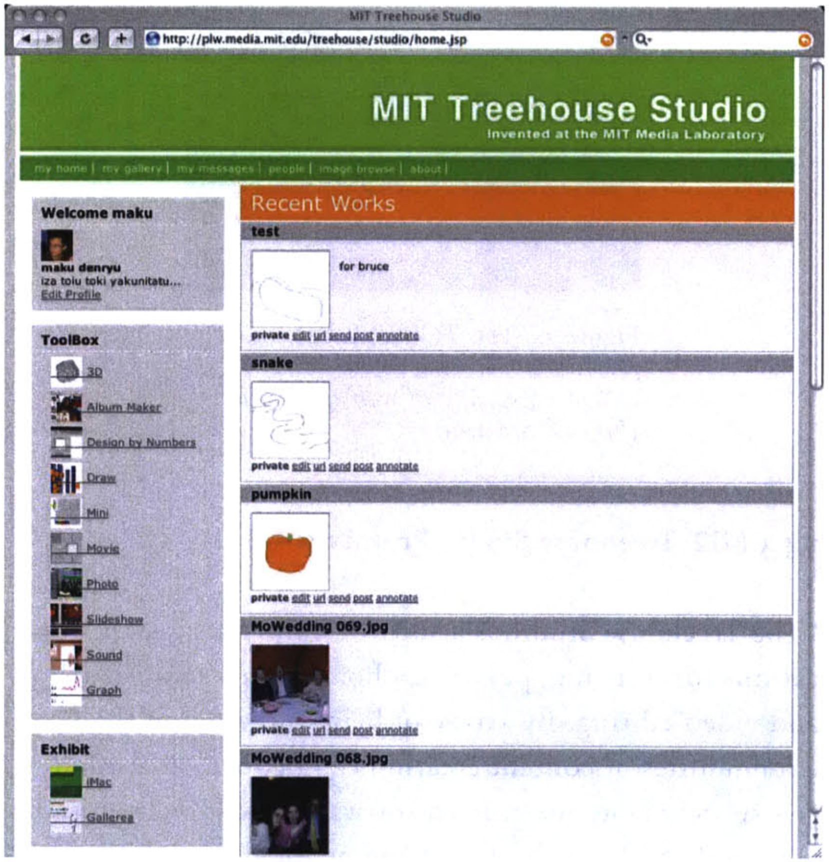


Figure 23. Treehouse Studio user home page

Built with Steamy at it's core, the Treehouse Studio Prototype. The tightly coupled design tools allow a wide variety of artists to generate content. After a user logs in, they enter their home page (Figure 18). On this screen, the user immediately sees the recent work in their portfolio, as well as a toolbox for digital design software. The Tools include a 3D modeler, a photo album maker, *Design By Numbers* [Maeda 1999], a vector based drawing program, a video editor, a presentation tool, and a sound editor.

All digital assets created with the tools are stored through Streamy to a centralized database. Publishing a work is as simple as changing a tag from private to public. Assets can be sent from member to member through a built-in messaging system.

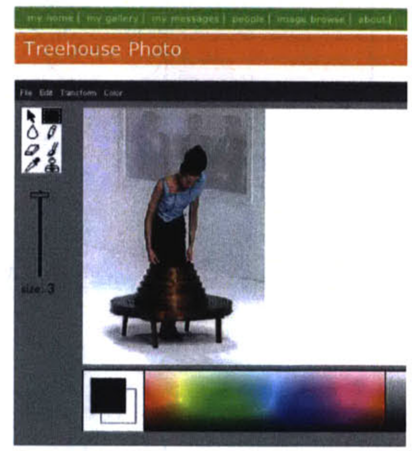


Figure 24. Treehouse Studio Photo Applet

2.4.4 Comm-piler

The "Comm-piler" is a web-based Java compiler. The initiative was to enable easy addition of new software to the system. Users would type text into a text area and then compile and launch a new applet. The advantage of such a system is that the user doesn't need a compiler or a software development kit. All of the user's source code and the generated executable are stored in the database. If the user is developing a new applet, pressing the run button will open a new browser window with the freshly compiled applet running inside. The Comm-piler can also be used to create on-the-fly plugins for web software such as the Treehouse Studio application. For example if you are using the Treehouse paint program, a savvy user could launch the Comm-piler winds and add new custom tools, filters and other software to the program.

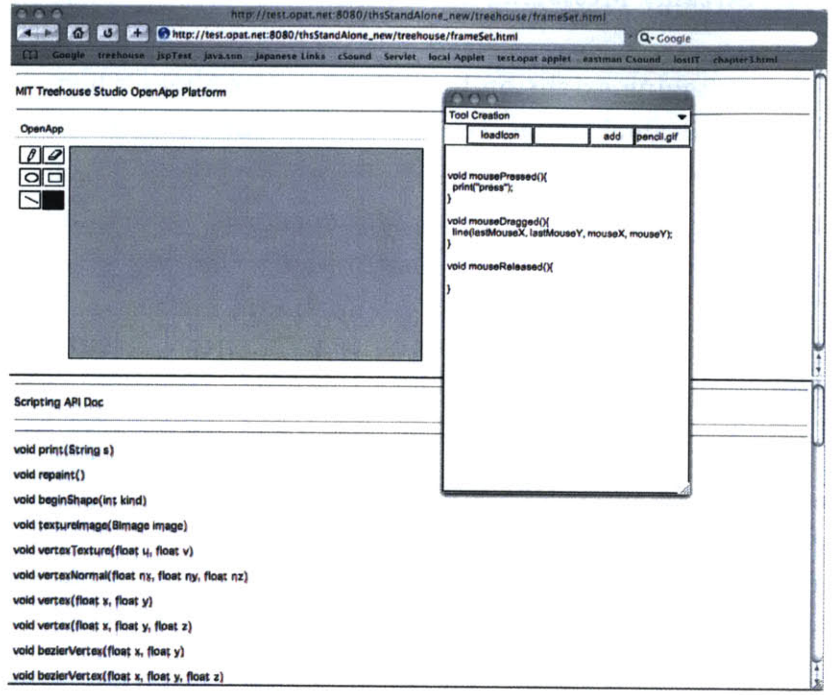


Figure 26. This experiment uses the Comm-piler to generate a tool on the fly.

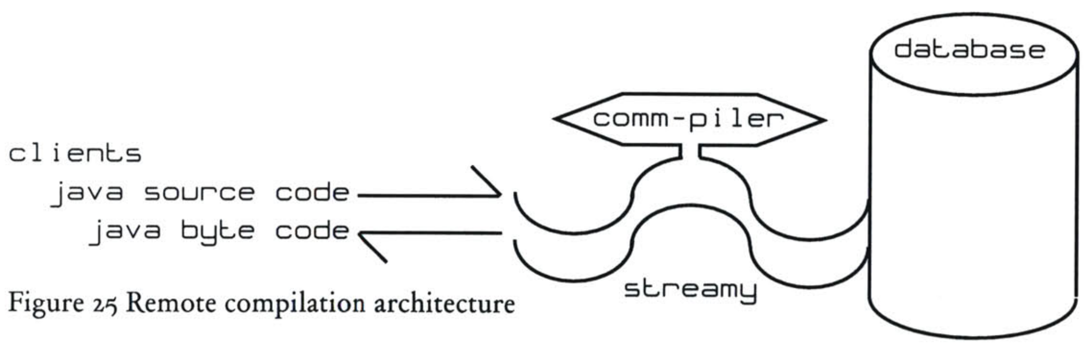


Figure 25 Remote compilation architecture

2.4.5 Streamy Messaging Extension

The messaging extension for Streamy came about when we needed synchronous communication for the system. The extension worked similarly to other messaging technologies such as *AOL Instant Messenger* and *Jabber*, but implemented a slightly different initialization procedure. When started, the messaging client tries to make a socket connection with the server. If this fails due to a fire wall, or some other restriction, the system defaults to HTTP polling mode. The polling mode is less efficient than using a dedicated socket connection, but has the benefit of using the http port 80, which is rarely fire-walled. Besides speed and efficiency, the user does not know the difference between the two connection modes. Alternatively, the Streamy messenger can negotiate and connect two clients for peer-to-peer style communication.

2.4.6 Draw Together

"Draw Together" (Figure 27) was software written and implemented using Streamy and the Messaging Extension. Two or more users could log in and draw interactively on screen. A List of the interaction would be kept. This is the first experiment where interaction history was stored with the document. Interaction can be played back like a movie.

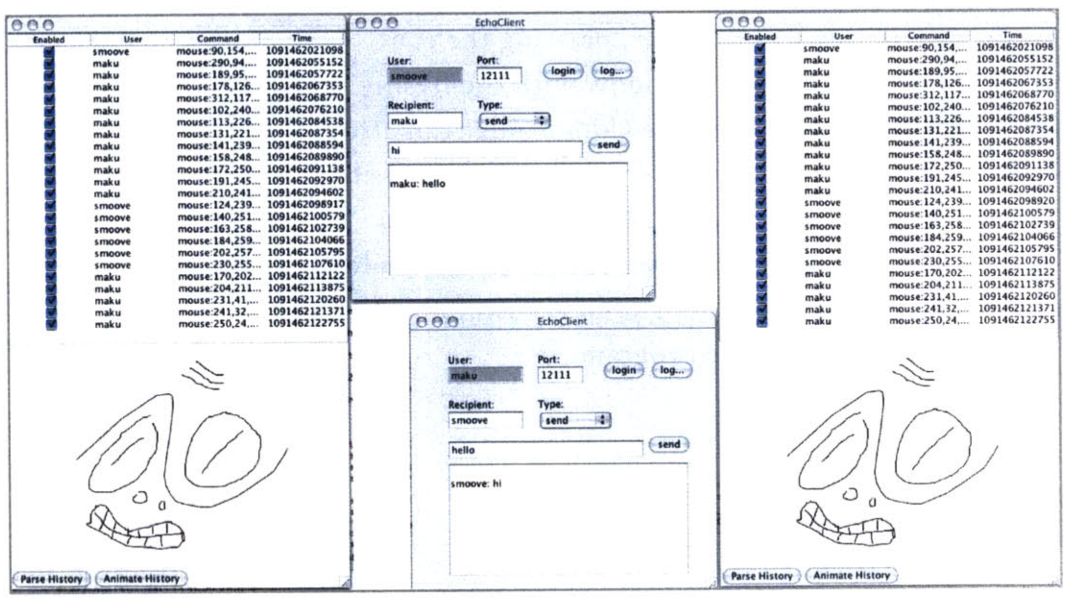


Figure 27. The Draw Together System makes use of synchronos communication and process tracking.

2.4.7 Code Together

"Code Together" was implemented using the same platform as Draw Together, but also used the network-based compiler, Comm-piler. Inspired by "Extreme Programming," [Beck 1999] the system aims to foster collaboration. The server keeps a database filled with source code, binaries, and lists of public methods and variables for each class.

The class-hierarchy map looks like a traditional class diagram, but contains a populous of avatars representing various programmers involved with different classes within the project. Clicking a class brings up the Togetheditor.

The Togetheditor adds chat room functionality to a traditional programmer's editor. Programming tasks can be created and assigned to a specific developer. As specified in Extreme Programming, one author edits a source file while another reads documentation, does research, comments the code and supports the coder. After a programmer designates a task as finished, the next person in the queue assumes control of the editor.

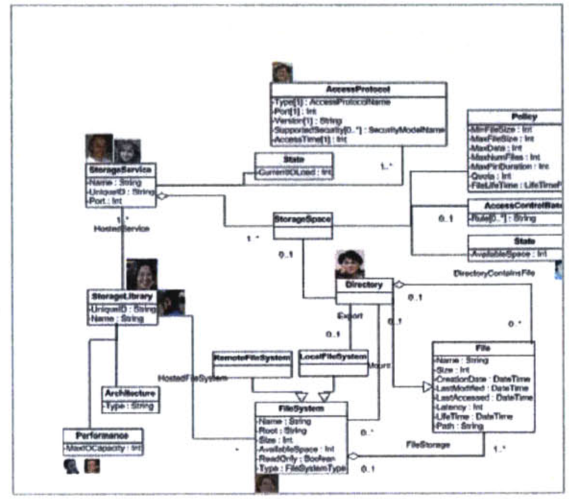


Figure 28. The Hierarchy Map

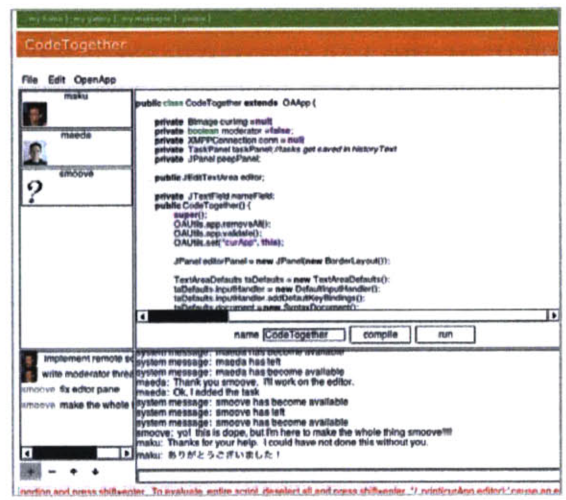


Figure 29. The Togetheditor

2.4.8 Record and Takeover

"Record and Takeover" involves capturing the user's input events by storing all mouse and keyboard events as they are appended to the Operating System's native event queue. The user can record their actions for as long as they like and then someone else can watch the computer repeat the user sequence as many times as they would like. One advantage of this system being implemented at the Operating System level was that it could be tested on a wide variety of software applications.

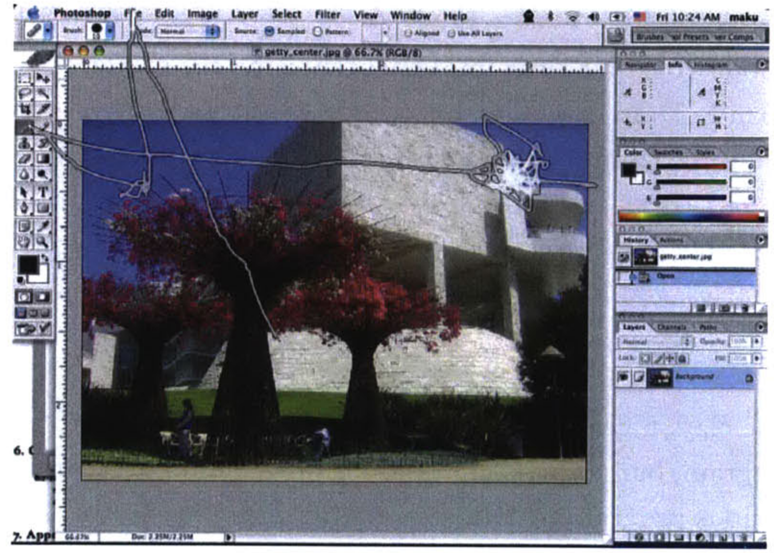


Figure 30. Using Record and Takeover with *Adobe Photoshop* on *OS X*. User loads an image, selects *Healing Brush*, and edits a portion of image. Plot overlay recreates path of mouse. gray path shows the area traversed by mouse with no buttons pressed; white area shows the area traversed with the mouse down.

Unfortunately, the playback software and the application communicate through the system event queue, so the viewer would manually have to return the application to it's original state after playback. Another idiosyncrasy of this system was that in playback mode, the software would completely take over the computer so the viewer could not interrupt or interact.
Testers of this system found it very surprising to see the system cursor control itself.

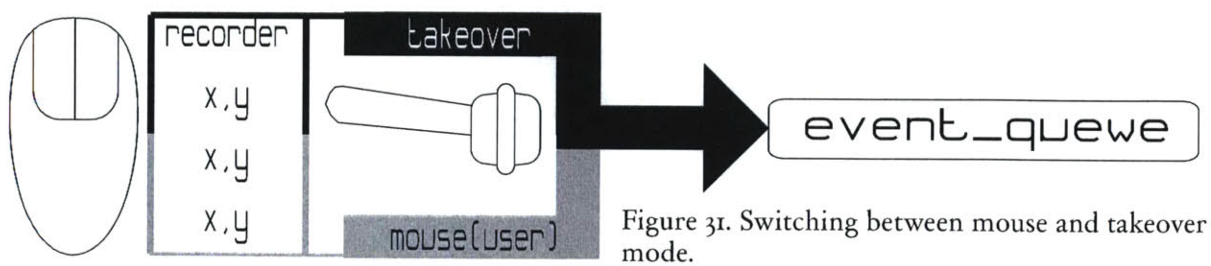


Figure 31. Switching between mouse and takeover mode.

2.4.9 SMPL

SMPL is a platform for rapidly creating web services. As web services become more important to application environments, it will also be more important to develop web services rapidly so the demand of delivering more web applications can be met. The SMPL architecture was designed with the shortcomings of streamy in mind. Although Streamy was an extremely simple system in it's implementation, it's simplicity made it's software very difficult to debug. When using Streamy, many errors could not be caught at compile time and only became apparent at runtime. SMPL aims to solve this problem and many others.

"It proposes an architecture and framework for the design and development of collaboration-oriented, distributed applications over the Internet. This framework... proposes an abstraction of the Internet as a network that is composed of services, resources, and capabilities instead of just machines. The goal of SMPL is to enable the development of applications that easily integrate the capabilities of different types of computing resources, software platforms, and data repositories across the Internet transcending the level of a single device. The SMPL architecture distributes resources through a peer-to-peer network of services providers. The design of SMPL encourages developers to add value to the system by facilitating the creation of new functionalities based upon compositions of the existing on" [Rocha 2005].

2.4.10 Toward Treehouse 1.0

As the original Treehouse studio prototype was a useful proof of concept, it was important to take another pass and try to meet some more goals with the application. Using the SMPL architecture at its core, Treehouse programs have a new potential to be powerful web applications. The top priority in the second release of the programs was to redesign the application API for extensibility in order to make it easier to develop new tools.

Figure 32. The Treehouse Studio Paint Program

3. Treehouse Historian

The Treehouse Historian is a reference implementation for process driven design software. The package gives the user the ability to annotate, search and play back the process of any work created in the system.

3.1 Example Scenario

The Artist starts by using the mouse to select drawing tools and sketch on the canvas. Instead of saving their file, they press the spacebar whenever they come to a pause in their process. After pressing space, a star appears in the corner of the screen to indicate a key-frame has been marked for later annotation (Figure 33). Alternatively, an annotation can be typed immediately by selecting the annotation tab, typing an annotation and pressing the new button (Figure 34). At any time, the artist may modify annotations and store them by pressing the edit button. One can get a birds eye view of the creation process by pressing the thumbs button and thumbnail images of all key-frames will be displayed. Clicking a thumbnail recalls and parses that stage of the process and puts the program in the same state (colors, selections, etc.) when the frame was keyed. The artist may now continue editing the file,

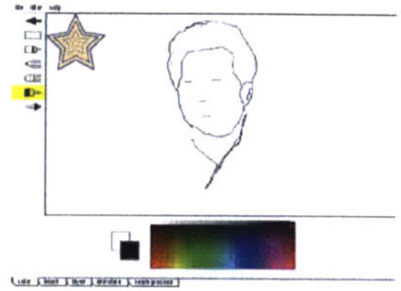


Figure 33. Using space bar to create a key frame gives the user a star as reward.

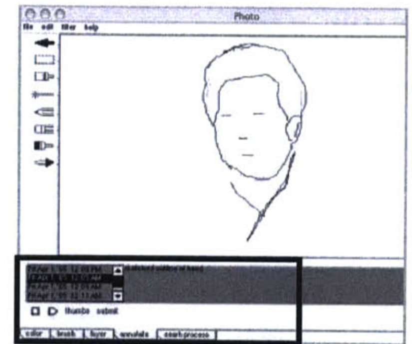
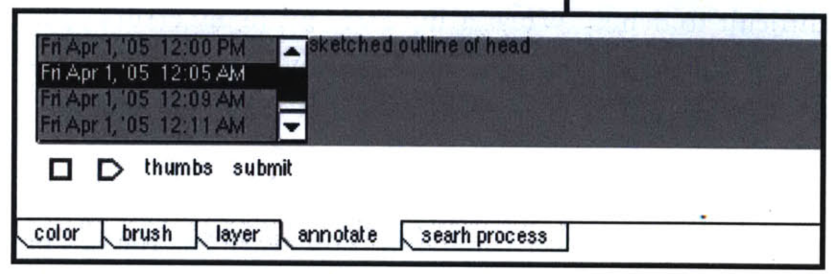



Figure 34. Treehouse Historian annotation panel allows process annotation at any point in the design cycle. Artists may annotate their work durring initial creation, or later durring playback. After submiting an annotation, the text is tagged with the timestamp of the last gesture compiled by the system. In addition, the client application may attach any metadata to facilitate specialized searches. In the case of the photo application, any gestures made on the canvas will attach the current tool, color, and brush size. All of this information buffers within the client application and then streams to the server where all of the data is stored.

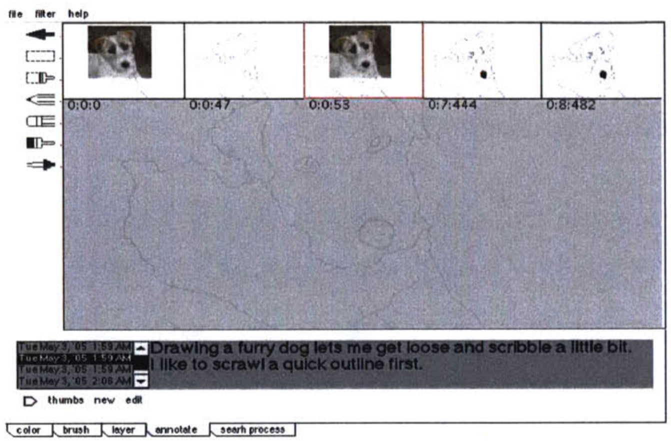


Figure 35 Bird's-eye-view of a Historian session

make additional annotations, or playback the process by pressing the play button.

Pressing the play button moves the cursor and shows the actual mouse gestures used to create the piece. All interactions are played back at the same speed as they were made. By default, the playback elapses until the next annotation is reached and then stops. Playback shows all interactions including menu, tab and button selection. Anytime during playback, the artist has the option to stop and make additional annotations.

Figure 36 Historian playback feature

The search panel serves as the interface for navigating all of the drawings in the system and allows users to query the process database for new techniques. After typing text into the search field, the user will immediately be presented with thumbnails of any files that contain annotations or metadata that match search criteria. For example, typing "eye" will bring up any works that have annotations containing that word. Selectable thumbnails appear and, as with the key-frame view, selecting a thumbnail allows exploration of that part of the process and gives the user access to annotation and playback. Alternatively, the artist may edit the drawing. In this case, the original data will not be overwritten.

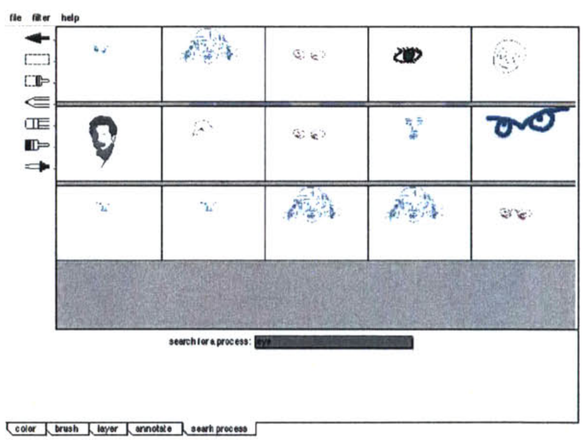


Figure 37. Historian search feature

3.2 Implementation Details

3.2.1 Gesture Recording

The Treehouse Paint program was designed and implemented prior to the implementation of this thesis work.

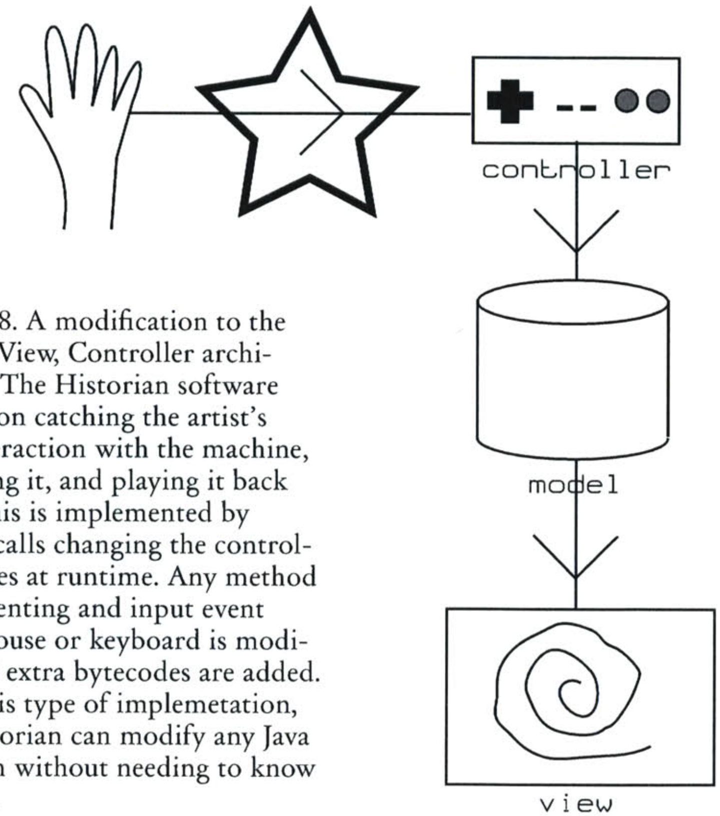


Figure 38. A modification to the Model, View, Controller architecture. The Historian software focuses on catching the artist's raw interaction with the machine, recording it, and playing it back later. This is implemented by dynamicalls changing the controller classes at runtime. Any method implementing and input event from mouse or keyboard is modified and extra bytecodes are added. With this type of implemetation, the Historian can modify any Java program without needing to know it's API.

The Paint program has a previously defined API and inherits from the Treehouse Client API. No modifications to the Treehouse Paint API have been made in the design of this thesis implementation. The Historian package, analyses the classes through reflection and modifies the byte codes of the Paint program at runtime. This byte-code manipulation occurs during class-loading. The procedure can be defined as:

```
Classloader
    loadClass(class)
        if(class implements a MouseListener){
            modifyMethod(mousePressed);
            modifyMethod(mouseReleased);
        }
        if(class implements a MouseMotionListener){
            modifyMethod(mouseMoved);
            modifyMethod(mouseDragged);
        }
```

The actual code modification was done using *Apache's* Java byte code engineering library, "BCEL." The modifyMethod function called here on all required MouseListener class types, adds three extra procedures to each of the methods in this example. The first procedure generates a time stamp of the call. The second creates a signature of the mouse event, including the X,Y coordinates, the event source and several other pieces of data such as mouse button flag. The third sends both pieces of information to The Historian.

3.2.2 The Historian

The Historian module manages the caching of events during recording, and the streaming of events to the server. The database mappings for gestures are:

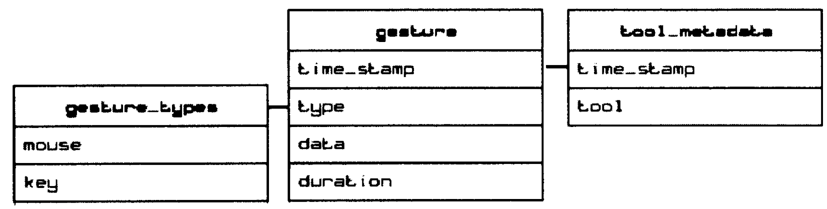


Figure 39 Historian database structure

Figure 39 shows how metadata can be attached to gestures. A gesture is a collection of input events from either mouse or keyboard. The Historian converts raw events into the Historian Event type and handles all client-side interaction, data compression and decompression. The Historian also handles the buffering of incoming events from the server and the dispatching of events during playback.

3.2.3 Historian Event Type

The Historian Event Type, or HEvent, is the object used to store each event. HEvents are different from regular Java Events, which must be sent to a dispatcher before they can be consumed. HEvents have the ability to execute themselves. This closely resembles the Command Pattern described in Design Patterns [Gamma, Helm, Johnson, Vlissides 1995]. HEvents contain coordinates, event source references, key codes, modifier fields, and appropriate time stamps. A collection of HEvents can be stringed together and processed as a stream. This capability facilitates the Historian Playback Feature. The structure of key and mouse HEvents are defined as:

```
HKeyPress {
    timestamp
    keycode
    ui_event_source
    modifiers
    durration
    execute()
}
```

```
HMousePress {
    timestamp
    x,y coordinates
    ui_event_source
    modifiers
    durration
    execute()
}
```

During real-time playback, each HEvent will subtract the previous event time stamp from its own and sleep for the resulting duration. This technique ensures that the playback precisely mimics the original gesture.

3.2.4 Process Annotation

A separate class, named AnnotationManager, handles the creation, editing, and deletion of anntaotions. This class also sends and recieves the associated text and time stamps to the database for storage.

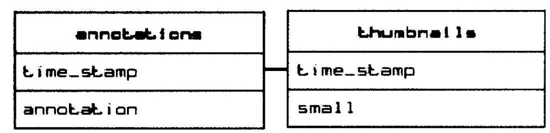


Figure 40. Annotations and thumbnails are also linked by times tamp.

3.2.5 Process Search and Retrieval

Searches are performed by querying the database for annotation key words. This is as simple as sending an SQL request to the server. Thumbnail images and time stamps for all hit are then returned. When a user selects the thumbnail of the data they wish to view, the associated time stamp is sent back to the server and the full data is retrieved. A socket is opened in the Historian and the interaction data is streamed in. As the data streams, the Historian parses the interaction data through the controller. This parsing occurs in exactly the same manner as playback, except their are no delays between HEvents. The system parses the data as fast as it can, rebuilding the painting from scratch. Benchmarking and optimizing this process was beyond the scope of the reference implementation, but the loding times were found to be acceptable by most users.

4. Evaluation and Analysis

4.1 What Can We Learn?

This section aims to evaluate the Historian system as a learning medium. Viewing some examples of works might provide an opportunity to see how people can learn and will serve as a method for evaluation the system. Here we will review three works just as a student might.

Three different works were selected because they show the application of different techniques. Two of these drawings are portraits, while the third is a still life. The first drawing was completed by a professional designer with much computer experience. The second drawing was done by a professional painter with little or no computer experience. They each took different amounts of time to complete. The third drawing was made by a computer programmer, who is very computer savvy, but has little art experience.

Skill Sets of Three Artists:

	Teaching	Artistic	Computer
#1	x	x	x
#2		x	
#3			x

4.1.1 Artist 1

Artist 1 drew a picture of a girl, made thirteen key frames, and annotated each.

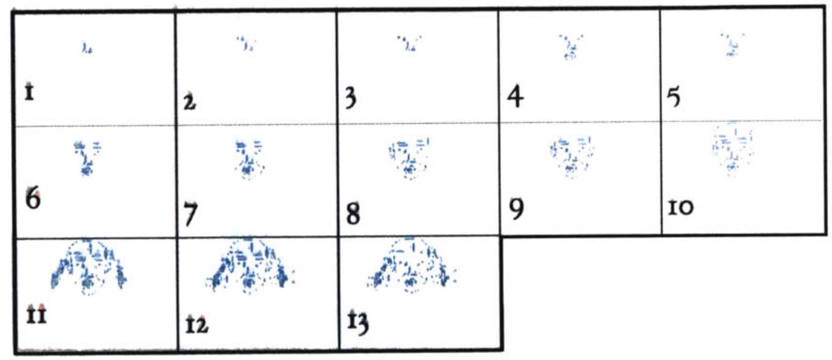


Figure 41. The thirteen annotated key frames made by Subject 1.

The corresponding annotation text:

1. Nose is an upside-down mushroom.

2. Eyes are like punctuation marks (think of quote marks)

3. Feminine eyes require longer eyelashes.

4. The mouth is hardest to draw often. it determines the emotions.

5. Drawing teeth is not so important, but it helps.

6. Eyebrows are also the other aspect of emotion-giving. Thick eyebrows are kind of silly.

7. Establishing the chin and cheeks is always done with a single line. don't draw multiple lines as it looks dirty.

8. An ear is something you can always fake. Just draw the outline, and them a random squiggle.

9. Do the same thing for the other ear.

10. Hair is best when you keep it simple. here she looks like a boy.

11. Pig tails however, make for the girl.

12. Bow ties make the pigtails seem even more girly.

13. Let's also be nice and give her more feminine eyebrows.

Figure 42. A still frame captured during process playback.

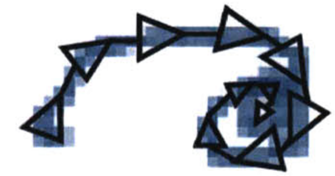


Figure 43. The arrows show the motion of the brush durring playback. While playing, the artists motor acitivy is imediately knowable.

Artist 1 does an excellent job of annotating his process, but zooming in and playing back reveals even more. Playing back Artist's 1's process reveals that the eye was created with one spiraling gesture (Figure 42).

In viewing the entire playback, we can see that many of the body parts were created with a single line. In this manner, we can see the style of the drawing as it is defined by the process of the artist. The final drawing (Figure 44) evolves through economy of motion and line.

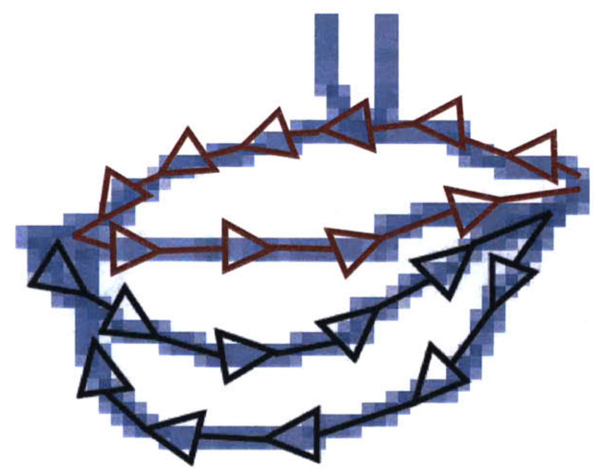


Figure 44. Artist 1 draws the mouth with 2 lines.

Figure 45. Artis 1's completed drawing.

4.1.2 Artist 2

Artist 2 is a professional artist who does not make use of digital tools in her work. She was given a laptop and brief instruction on using Treehouse Paint. She used the search feature and watched playback of several drawings in the system. A still life was set up from objects in the studio. She was asked to paint, make key frames and annotate as she saw fit.

Artist 2 drew for a longer duration than Artist 1, yet an-

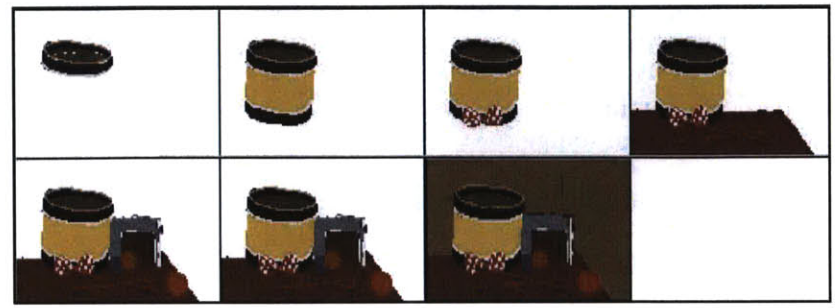


Figure 46. Artis 2's key frames.

Annotations:

1. Start drum. It's more or less a basic cylander. The most imprtant part is setting the lip.

2. Draw body of drum. Contours should complete the cylander.

3. Draw polka dot bow tie. Adding a few highlights helps to define the form.

4. Draw table and drum sticks. The thin sticks need some shadow

5. Draw the balls in the foreground.

6. Draw radio in a layer. Eraser was used to make it appear behind the balls.

7. Finished!!!

notated fewer key frames. The annotations she wrote were useful, but it is necessary to take a look at the process to see how she made the painting. Many stylistic differences exist also. Here, the technique involved exacting an outline though painting, erasing, and repainting many times. This sharply contrasts the economy used to create the previous drawing. The priority in the still life drawing was color selection and precision application of paint. The artist worked at the still life for over forty minutes, while Artist 1's drawing took just over two minutes.

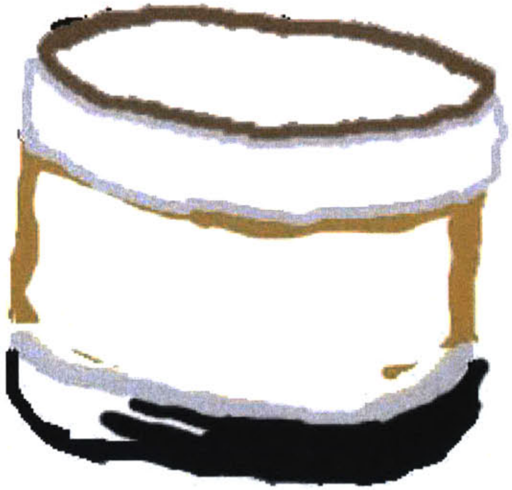


Figure 47. Artist 2 would select a color, work and rework the outline and then fill in the inside of outlined shape with color.

Figure 48. The Artist used layers and organized the objects fron to back on independent layers.

Figure 49. Artist 2's final painting of the still life

4.1.3 Artist 3

With little teaching or artistic experience, Artist 3 excels at anything involving a computer. He required very little instruction because he learned experientially by quickly trying all of the software features.

This artist did mention that they were tracing in the first annotation, but it's difficult to get a sense of the process

Figure 50. Artist 3's key frames. This Artist traced over an imported photograph and toggled it's visiblity during the process.

Annotations:

1. Begin tracing head.

2. Trying to blend all of the areas of color.

3. This sucks, DO OVER!

4. Let's go a different route on this. Let's make a yellow outline.

5. Outline the entire figure in yellow.

6. Fill in black hair underneath.

7. Do the watch.

8. Paint in skin tone.

9. Add highlight on skin.

10. Remove trace image and outline for good.

from the annotation alone. Playing back various sequences of this drawing reveals a complex technique. The artist begins by selecting colors from the tracing photograph and applies them in a painterly way on a transparent overlay layer. After about two minutes of experimenting with the technique, he switches strategies completely. Choosing yellow, which stands out from the tracing image, he traces an outline over the entire image. Then he creates a new trans-

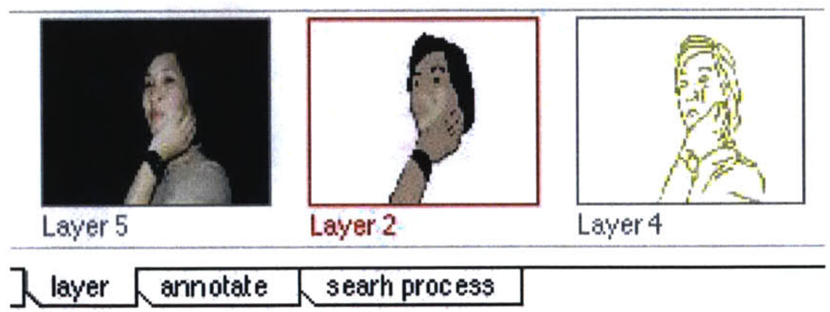


Figure 51. The Layer Panel shows the three layers involved in Artist 3's drawing.

parent layer in between the photograph and the yellow outline. He then uses this layer to paint in solid blocks of color. Once again, he uses the technique of using the *eye dropper* tool to select actual colors from the photograph. He breaks this convention by selecting the hot pink shirt color on his own. Once the colors were filled in, the artist deleted both the photograph and the yellow outline layers, leaving only the middle layer.

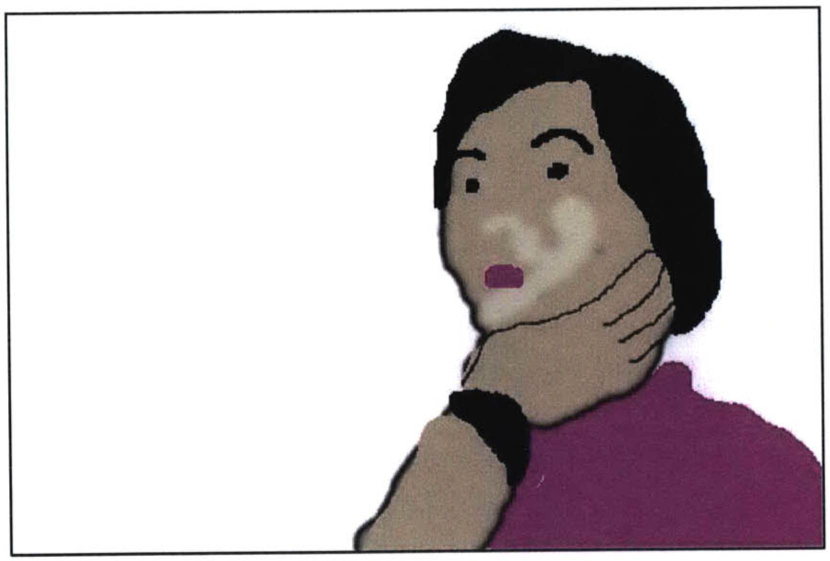


Figure 52. Artist 3's final drawing

4.1.4 Conclusions

Here we have presented the works of three very different artists with very different backgrounds and skills. They have created and annotated drawings. We viewed the playback of each drawing and analyzed the technique. It is important to note that the annotation style can differ as much as the artistic style. For example, much variance exists in the frequency and length of the annotations in the three works.

Drawing 1:	
Session time	2:21
Number of annotations	13
Median length of annotations (words)	10.8
Percent of session time spent annotating	%43.7
Drawing 2:	
Session time	42:33
Number of annotations	7
Median length of annotations (words)	9.7
Percent of session time spent annotating	%6.5

Drawing 2:

Session time	17:11
Number of annotations	10.0
Median length of annotations (words) 5.70	
Percent of session time spent annotating	%7.16

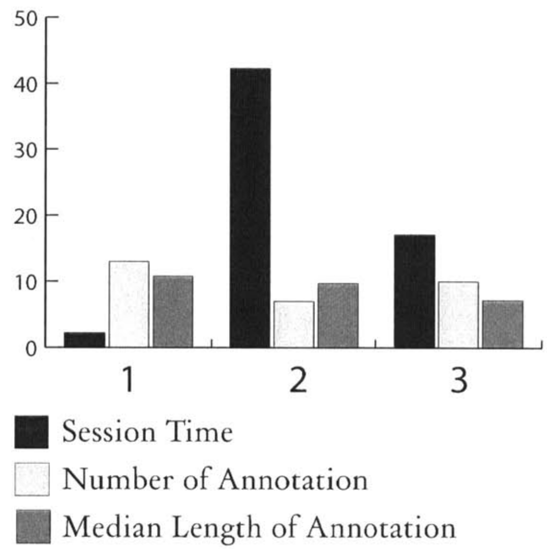


Figure 53. Annotation-style comparison

The users annotation patterns and habbits were all quite different. Suprisingly, the artist who drew the fastest also wrote the most. Sampling a larger set of drawings might give a btter picture of the annotation process.

4.2 Further Analysis of Annotation Process

Several volunteers were chosen and invited to sit at a personal computer. As a preliminarily, they received a brief demonstration of the paint software and annotation system. After a few minutes to experiment with the tools on their own, they were asked to perform the following specific task:

1. Please draw a face in any style.

2. Create annotations using either of these methods:
   a. Press space bar during natural pauses in your process to create key-frames for later annotation.
   b. Press annotation tab and annotate as you create the work.

3. When the drawing is complete, please annotate any remaining key frames.

4. Review your process. Make any necessary corrections and feel free to add annotations.

After the drawing and annotation session, the volunteers were invited to search through the database. Their inter-

actions were observed and noted. An informal interview followed each session and feedback was recorded.
By the end of this experiment, twenty-three faces were drawn by seven different artists. A program tabulated quantitative statistics. Highest and lowest values were discarded for medians.

4.2.1 Data Summary

Mean session time per face (min.)	5.3
Mean annotations per drawing	7.0
Median length of annotation (words)	4.9
Percent of session time spent annotating	%35.2

At first glance, the percentage of time spent annotating seems exorbitant, but the percentage changes for longer and more complex drawings. A search into the database for all drawings made during longer sessions revealed a different statistical profile. Querying for drawings with session times between twenty five and thirty minutes (six drawings) produced the following results:

Mean session time per drawing (min.)	27.4
Mean annotations per drawing	12.0
Median length of annotation (words)	18.6
Percent of session time spent annotating	%12.3

Comparing the two sample sets, 5.2 times more time elapsed in the longer studies, but the number of annotations only increased by a factor of 1.7. The median length of the annotations increased by a factor of 3.8. The shorter studies; however, showed a %285 increase in the fraction of the total session time spent annotating.

Expanding these observations out to the entire sample set results plots the curves shown in Figures 54, 55 and 56. Each of the three diagrams represent findings for all 75 drawings currently in the database. They are scatter-plots with 5th degree polynomial curve-fitting applied.

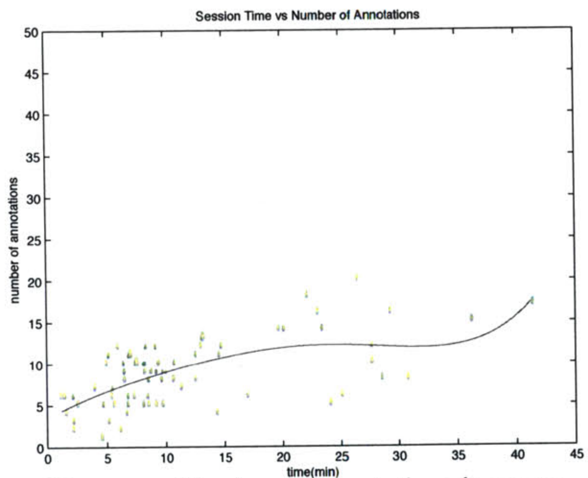


Figure 54. Number of annotations increases slowly as session duration increases.

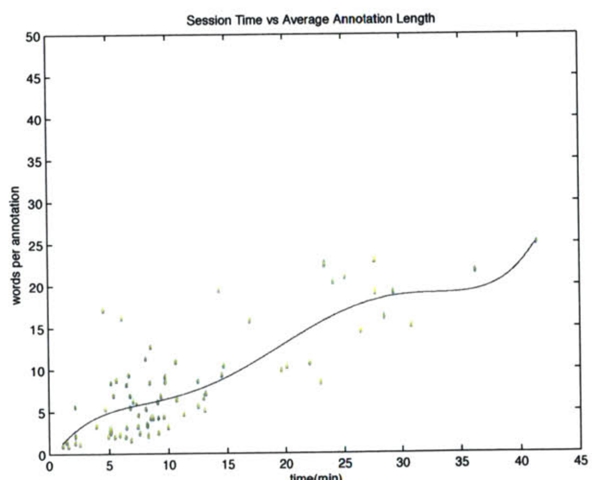


Figure 55. Length of annotations increases sharply as session duration increases.

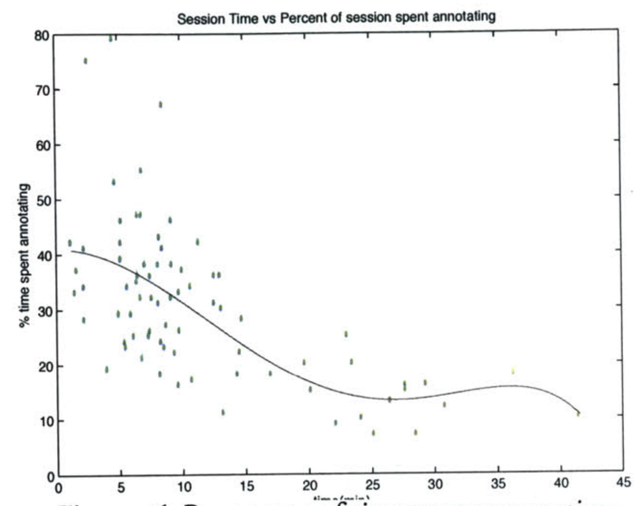


Figure 56. Percentage of time spent annotating declines sharply as session duration increases.

Figure 54 plots the total session length (drawing and annotation) on the horizontal axis, and the number of annotations on the vertical. The linear slope of the data is .2. Fiure 55 plots the time (horizontal) and words per annotation (vertical.) The linear slope is .4. An interpretation of these slopes tell us that in the current system, artist write longer annotations for more time intensive works, but don't make proportionaly more keyframes. Figure 55, which plots session time (horizontal) and percent of that session time spent annotating (vertical,) has a linear slope of negative .5. This plot shows that eventhough longer works require more numerous and much longer annotations, the proportion of the artist's time spent annotating declines sharply as they work longer sessions.

4.2.2 Conclusions

Introducing annotation into an application always creates more work for the user. In this experiment, the ratio of time devoted to annotation declines as the work session becomes more lengthy. The key metric studied here is the percent of time spent annotating, which should be of principle concern to anyone designing an annotation system. When ever possible, user interfaces should be designed to streamline the process of annotation. This will greatly reduce the burden on the user.

4.3 Functional Comparisons

Section 2.1 introduced The Theory of Multiple Intelligence and presented several different learning styles. Section 2.2 presented several different learning-software precedents currently in use today and discussed them in the context of the learning style they most closely modeled. This section will compare the suite of learning tools in Treehouse Historian to some of the more relevant software precedents.

4.3.1 The Intelligences of Treehouse Historian

The Theory of Multiple Intelligences, although typically applied to people, may provide an interesting metric to evaluate software. This sort of analysis considers how an artist's intelligences converge when using the software. A paint system undoubtedly engages Visual Intelligence. The artist must consider their work as a whole. Details must be adressed, but they are considered by their affect the entire picture. Digital painting also activates the kinetic mind because the artist must make gestures with a stylus or mouse. Connecting the system to "Tangible Interfaces" such as the "IOBrush" [Ryokai, Marti and Ishii 2004] could make the experience even more kinetic.

When the artist stops and annotates, the Verbal and Logical Intelligences begin to interact with the Visual. The artist must consider the details of the process at discreet keyframes. They mustreflect (Intrapersonal Intelligence) on the process and describe the steps. In adition, the Artist must determine the best way to communicate their activity through annotation, which engages Interpersonal Intelligence.

While reviewing a process, the artist first uses logic to select proper search criteia and successive processing as they incrament through the steps. The thumbnail views aim to provide a "bird's eye view" in order to re-engage simultanious processing. The playback feature aims to describe the motor activity of the artist and give kinetic information to the reviewer.

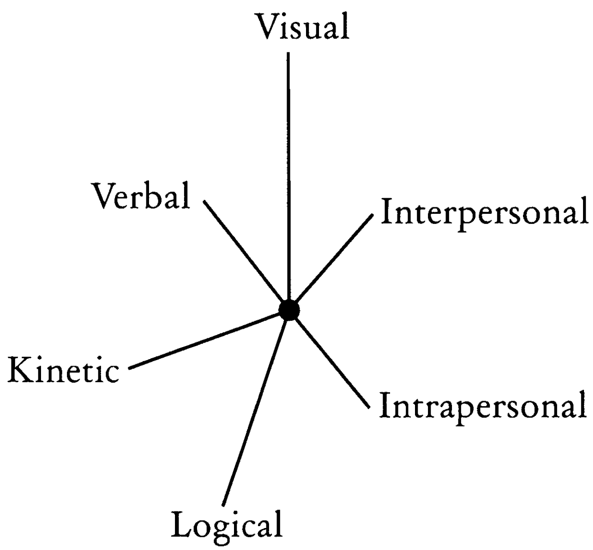


Figure 57. The Multiple Intelligence distribution of Treehouse Historian.

4.3.2 Recipes vs. Process Annotation System

The sequential nature of tutorial recipes follow a linear narrative pattern make them very applicable to logical and verbal learning styles. Recipes closely follow the model of left-brain successive processing because the information accumulates from sets of details. Following it may not be appropriate as the primary format for delivering art and design tutorials where a visual and kinetic learning style is typically more appropriate. Right-brain dominant simultaneous processors, however lack the ability to get a "bird's eye view" of the process. Recipes, however, do offer a good example of how to linearly annotate a process and therefore can be used to comparatively evaluate the annotation functionality of The Process Historian.

Using the Historian's annotation panel, a successive processor may step through the process in a linear manner. Additionally, thumbnails can be displayed at regular time intervals to display a general overview of the process. Global or holistic learners may find this preferable to the lockstep nature of annotation-only tutorial recipes.

Typically, to make a tutorial recipe, one must rehash a previous work. In doing so, the process being described is a reenactment. During the reenactment, the author must stop periodically and explicitly perform a screen capture. This capture must then be formatted and accompanied with instructions and other notes. If the author forgets to do a screen capture for an important part of the project, they must recreate that portion. To create a tutorial using the Historian, the author need not contrive or recreate any scenarios. The process of creating a tutorial is implicit. Annotation is the only extra step needed and may take place in realtime or after the fact.

Web search-ability is one advantage that lockstep tutorial recipes do have over the Treehouse Historian. As HTML serves as the de facto standard file format for tutorial materials, search engines can be used effectively. Conversely, Historian defines it's own data structure and is currently limitatited becasue processes are only searchable within a Treehouse application.

4.3.3 Tutorial Video vs. Process Playback

The Treehouse Historian process playback system might seem very similar to annotated tutorial videos on the surface. They both emphasize a visual and kinetic learning style, but the real difference is in their design.

Tutorial videos, generally created using motion screen capture software, sample the frame-buffer of the display device at a given rate. A single snapshot of a 1024 by 768 pixel screen will require an integer array with 786,432 indexes to buffer all of the data, and large amounts of computation to compress the data into a suitable video format. Requiring considerable computation power and system memory restricts the capabilities of the machine being recorded. Special video equipment must be used to circumvent this problem. In either case, it is simply not feasible to keep capture software running all of the time. Since one must explicitly start and stop recording, tutorial videos can seem contrived.

Bytes Required to buffer:

- 1024 · 768 screen capture = 3145755
- mouse event = 328

Conversely, the process playback system designed here does not grab pixels or make use of a video codex. Input events from mouse and keyboard are played back through the system. The computational requirements needed to sample a mouse event are minimal when compared to sampling the display buffer. This fact makes it possible to run the capture software implicitly in the design tool.

Playback will always be possible for every document, not just the chosen ones.

Choosing an event-driven playback design emphasized the priorities of the Treehouse Historian system, but a design using captured pixels and a compressed video codex has some advantages. The event-driven system limits how the playback can be viewed. The playback of the reference can not be slowed down, sped up, or reviewed backwards. In addition, making use of a popular video codex allows the learning materials to be off-loaded to personal devices such as PDAs and Smart-phones.

One interesting opportunity arises in a hybrid scenario screen takeover described in section 2.3.4 and the tutorial video scenario described in section . In this model, the user would not need a quick eye and a strong memory to learn for the system administrator takeover sessions. Videos could be automatically be made and the learner could review them any time they wish. This scenario could be further improved if the system administrator's voice could annotate the session while performing the task on the learners computer. These annotations would be a useful supplement to the video and better than the video alone. The video could then be presented online for others to see. That way a one-on-one tutoring session could then benefit many as it is likely that others will encounter similar situation.

The above scenario might be applicable to the Technical Support and System Administrator space, where there is mutual incentive for the teacher and learner. The System Administrator has clear incentive to annotate the help sessions because it will decrease their work load over time as the learner will require less repeated support.

4.4 Feedback

As the software presented is largely experiential, obtaining a round of user feedback proved very necessary. Feedback was gathered in a variety of venues. First, during demos at the MIT Media Lab. Second, during a two-day Simplicity Consortium event on Cape Cod in late March, 2005. Third, the trial summarized in Section 4.1. I acknowledge that this is not an extensive trial, but provided enough preliminary data to better evaluate the corpus of work. This qualitative data will provide valuable in considering future improvements and to further contextualize the work. The following subsections will provide a summary of this qualitative data.

4.4.1 Treehouse Paint Tool

Since the process tools were tightly integrated with the paint tool, it was often difficult to ask solely for feedback regarding the Process Historian. In fact, most of the feedback I received dealt with the usability of Treehouse Paint.

The interface of paint was found to be simple and easy to use. Experienced computer users found the interface to be instantly knowable, while novices could learn in minimal time. Many users commented that the tabbed interface on the bottom of the screen was novel. One user suggested that the tabbed pane be splitable so more than one tab could be seen at a time. Another suggested that the tabs should appear and disappear off of the screen to conserve space.

As for the tabs themselves, One user suggested that color selection tab should allow for loading and mixing additional colors. Another suggested that brushes should be available in shapes other than circle. Users experienced with painting and photo editing software found the layer panel very usable without instruction, while users who were unfamiliar to the concept of layers required some instruction.

Many users found the number of tools satisfactory. The main tool missed was the rubber stamp, which is available in most major paint and photo applications. Users found the Selection Brush novel and interesting, while the Paintbrush felt less comfortable than those in commercial paint software.

4.4.2 Process Annotation

The annotation system was a cause of concern for many users. Generally, they thought that people would not annotate sufficiently to make the system work. One user said that the system would work great in a for-profit design environment where employees could be expected to annotate as part of their job description, but it would otherwise be difficult to get people to annotate. Conversely, the recording of metadata such as the type of tool being used was found very useful. Some users suggested that more metadata be used by the system.

4.4.3 Playback

The Record and Takeover experiment described in Section 3.2.3 held significant wow factor with users as the automated system cursor would fly around the screen during playback. The Treehouse Historian did not takeover the system in the same totalitarian manner, so the system sursor did not move. Users found this frustrating as they didn't always know where to focus their attention. In addition, many users wished that the playback sequences could be played back slower, faster, or in reverse.

4.4.4 Search

Overall, users were impressed with the speed at which the search mechanism displayed thumbnails and wondered if this would still be the case if the database were very large. At a first glance they were excited by the prospect of searching data by date and time, and once the data was loaded, being able to go back and forward in time. As annotating and searching by process was a completely new concept for many users, they were not always comfortable with this interface at first. After typing in several queries, the system became more comfortable and understandable.

5. Conclusion

This thesis has explored an architecture for developing design tools that record the artist's process. This exploration was done with the intention of creating a new tutorial platform specific to learning art and design software techniques. A brief study of cognitive learning styles has helped to evaluate both currently available tutorial software platforms and the new system presented in this work.

A preliminary evaluation has been performed but a major question remains: do artists want to share their process? Some artists may consider their technique trade secret and feel they give up competitive advantage by sharing. While security and file permission were not part for the reference design or implementation, such a system could be used to allow artists to broker their own processes.

5.1 Future Directions

Using interaction as the native data structure offers many opportunities outside the realm of this body of work. The three scenarios that follow represent some interesting paths found during the development of this corpus of work.

5.1.1 The Danger Mouse Scenario

In 2004, DJ DangerMouse released the *Grey Album*. This work was entirely comprised of samples from Jay-Z's The *Black Album* and *The White Album* by The Beetles. It is a wildly successful album was downloaded over a million times in its first week of release. Danger Mouse, the author, could not profit from the work as it was entirely illegal because the source materials were copywriten. Recording the process of creation in a digital arts scenario could open profit opportunities for artists like Danger Mouse because the process of creation itself could also be copywriten and sold. A perfectly legal way to distribute the Grey Album is to sell the process to listeners who already own both the Black and White Albums. The customer could simply apply the process themselves.

Figure 58. *The Grey Album.*

5.1.2 Focus Group Scenario

The developers of the creative software might also make use of the data. By reviewing the specific interactions with a piece of software, developers could get a better sense of how their code is being used. This analysis could then be used to create new features for the software or to remove ones that have become vestigial. This could solve some "code bloat" issues as developers and quality assurance engineers would have hard data to support the removal of a feature. In this manner, the interaction history can be harvested in a way that effectively replicates large scale focus groups.

5.1.3 UI Augmentation Scenario

Computerized searches of the interaction history can also take place. For example, if a user continually uses four commands in succession, a computer program could discover this and ask the user if he would like the sequence combined into a macro and added to the interface through a button, menu, or some other user interface component.

5.1.4 More Active Interfaces

The use of the reference implementation took place either on a laptop or desktop PC with a standard keyboard and either a mouse or tablet device. This served as the preferred platform for preliminary testing, but interfaces such as the "IOBrush" [Ryokai, Marti and Ishii 2004] would present many interesting opportunities. More expressive interfaces and tools, the more valuable the recorded process data becomes.

5.2 The Ninth Intelligence

Some students of Gardner's Multiple Intelligence Theory have sharply debated that a ninth intelligence exists. The ninth, "Existential Intelligence," deals with the ability to ponder deeper issues such as the meaning of life, why we are born, and why we die. One way of thinking the self is to ponder one's everyday actions. In addition to considering ones self, persons also look at the actions of others for inspiration. Conversely, sometimes the actions of others serve as blueprints of what not to do.

For the Artist, many existential questions exist. How do we define ourselves and our legacy? Should it be though achieving fame, wealth or some other kind of external symbol of success? This may work to motivate some people, but the most important thing is to focus on the creative task on hand and to concentrate on the specific details of the creative process. Innovation lies in the doing, and charity lies in the sharing. Teaching other people what we have learned in our lifetimes is our true legacy.

John Tchalenko's work with Humphrey Ocean is extremely important in that it propels artists and art educators to understand the creative process better. As Chapter 2 of this thesis explored learning styles, Tchalenko will continue to achieve a better understanding of creative styles. Tchalenko's experiments only include one artist, but could be expanded to thousands with web software. A simple web camera can very closely mimic the functionality of the sophisticated eye-tracker used in the Ocean experiments. The starting point is to be able to study the motor activity of the artist. If we can compare eye and hand movements, we can formulate theories about the effectiveness of various motor patterns.

In conclusion, practical applications exist for the technology presented here, but these specifics should not obscure the core concept. The best tool to learn from is the one which can record and share the manner in which it was used. Ancient civilizations left their tools behind. These artifacts taught us who they were. The creative legacy of our civilization will largely be digital. Having digital tools that keep track of our process will not only tell us about ourselves, but will inform generations to come.

Bibliography

Adobe Software <www.adobe.com>

The Apache Software Foundation<www.apache.org>

Apple Computer <www.apple.com>

D. S. Backer, *Structures and Interactivity of Media: A Prototype for the Electronic Book*. PhD Dissertation, Massachusetts Institute of Technology, 1988.

K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 1999

L. V. Berens, *Dynamics of Personality Type: Understanding and Applying Jung's Cognitive Processes*. Telios Publications, 1999

B. Edwards, *The New Drawing on the Right Side of the Brain*. Putnam Publishing Group, 1999

Famatech <http://www.famatech.com/>

E. Freedman, "Learning Styles, Culture & Hemispheric dominance." < http://www.mathpower.com/brain.htm > 2001

E. Gamma, R. Helm, R. Johnson & J. Vlissides, *Design Patterns*. Addison-Wesley Professional, 1995

H. Gardner, *Multiple Intelligences: The Theory in Practice*, Basic Books 1993

H. Gardner & T. Hatch, *Multiple Intelligences Go to School: Educational Implications of the Theory of Multiple Intelligences*. Educational Researcher, 1989.

S. Kumar, *Wink*. <http://www.debugmode.com/wink/> 2005

B. Leuf & W. Cunningham, *The Wiki Way*. Addison-Wesley, 2001.

B.M. Manner. "Learning styles and multiple intelligences in students: Getting the most out of your students' learning." *Journal of College Science Teaching*, 2001

B. McCarthy, *The 4MAT System : Teaching to Learning Styles with Right/Left Mode Techniques.* Excel, 1987.

Microsoft Corporation <www.microsoft.com>

My Janee <http://www.myjanee.com>

M. F. Paulsen, "Online Report on Pedagogical Techniques for Computer-Mediated Communication." <http://gaya.nki.no/~morten/cmcped.htm > 1995

Performance Learning Systems (plsweb), Careers Categorized by Intelligence <http://www.plsweb.com/resources/articles/mi/2004/03/17/> 2004

Photo Shop Techniques <www.photoshoptechniques.com>

F. Popper, *Art of the Electronic Age.* Thames and Hudson, 1993.

B. Prashnig, *The Power of Diversity: New Ways of Learning and Teaching.* Bateman 1998

Quick Click Training <http://www.quickclicktraining.com>

C. A. Rocha, *SMPL: A Network Architecture for Collaborative Services*, MS Thesis Proposal, Massachusetts Institute of Technology, 2005.

K. Ryokai, S. Marti and H Ishii, "I/O Brush: Drawing with Everyday Objects as Ink." *ACM Press*, 2004

Symantec Software <http://www.symantec.com/index.htm>

J. Tchalenko, Drawing and Cognition. <http://www.arts.ac.uk/research/drawing_cognition/john.htm> 1998

J. Tchalenko & C. Miall, "A Painter's Eye Movements: A Study of Eye and Hand Movement during Portrait Drawing." *Leorardo* 2001

Tools4Ever <http://www.tools4ever.com/products/utilities/adminmagic/>

F. Viégas, M. Wattenberg and K. Dave, "Studying Cooperation and Conflict between Authors with History Flow Visualizations." *ACM Press*, 2004

Wikipedia <www.wikipedia.com>