History and Programming

Eduardo Bellani *

December 2, 2010

Abstract

Those who cannot remember the past are condemned to repeat it. Because programmers usually think they deal with cutting edge technology, they tend to forget the age and genealogy of the ideas they are working with. A demonstration of the history of the some crucial ideas of the programming craft would avoid the repetition of error and allow better ideas to take hold.

keywords: Programming, History, Learning

1 Introduction

Almost every school that tries to form professionals(?) begins with an overview of the history of the field. That is doubly true (?) in art schools, because one is expected to learn to develop his own style by studying the style of others. Programming, which is usually seen as an art discipline as much as an engineering one, does not maintain that tradition. That leads to a lack of an historical perspective of the discipline methods, points of views and techniques.

A professional without an historical perspective could be compared to a navigator without a compass. The navigator can move, but hardly advance towards a goal if not by accident.

2 Ideas

Here I will approach three of the main ideas that integrated to form the modern personal computer environment: Graphical user interface, constructionism, and augmentative systems. Kay [1993]

*blog – contact

This division will at times seem artificial, not only because they influenced each other, but because there is great overlapping of techniques and philosophy between them. On the other hand this structure does capture important distinctions and points of view that are valuable to understand the whole story.

As with any history, there can be a great level of details, and all of the ideas presented here have much more intricacies and were implemented in a great deal of different systems. Most of this fine grained idiosyncrasies will be left out because of my intention to give a high level view of the subject. This subject was treated in a more detailed and authoritative view by the references used here, which the reader is encouraged to follow.

2.1 GUI

The development of all graphic environments can be traced directly to Ivan Sutherland's sketchpad. Its ideas *still influence how every computer user thinks about computing.*Sutherland [2003]

It did that by introducing graphic metaphors and devices, such as the light-pen, the rubber band graphic manipulation, icons and a great deal of the things most people take as "natural" when manipulating modern GUI. Sutherland [2003]

But the major influence for programming paradigms was because its object oriented features, which along with Simula [1] had a major impact in the inspiration and development of Smalltalk:

> What Simula was allocating were structures very much like the instances of Sketchpad. There were descriptions that acted like masters and they could create instances, each of which was an independent entity. What Sketchpad called masters and instances, Simula called activities and processes. ...
>
> This was the big hit, and I've not been the same since. Kay [1993]

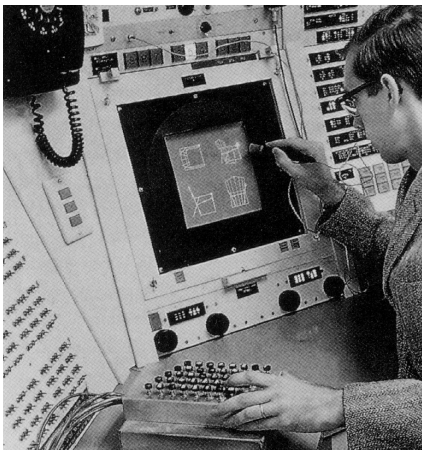

Figure 1: Sketchpad console, circa 1962. Müller-Prove [2002]

[1] Both share a common ancestor in the work of Douglas T. Ross Sutherland [2003]

2.2 Constructionism

Programming is an activity that deals primarily with the construction of abstract structures from ideas. This process and some of its encompassing theories are usually neglected or just fade into the conceptual background. This void is filled by constructionism, a learning theory developed by Seymour Papert and his colleagues that:

> ... shares constructivism's connotation of learning as "building knowledge structures" irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe. Papert and Harel [1991]

The proponents of this theory were the front-runners of using computers as a *personal media*. They did that in order to propose an alternative to instructionism and technocentrism. Those ideas can be summarized in the claim that *"To get better education, we must improve instruction. And if we're going to use computers, we'll make the computers do the instructon."*. Papert [1980]

Constructionism's alternative consists of using media, technology, and social environments as scaffolds to the conceptual structures that is to be built by people in the active role of their education. This education consisted initially of children learning math supported by constructionism's answer to traditional math curricula, Logo.

Logo profoundly influenced modern computer languages and environments through the heavy impact it had on the development of the Smalltalk system:

> I finally visited Seymour Papert, Wally Feurzig, Cynthia Solomon and some of the other original researchers who had built LOGO and were using it with children in the Lexington schools. Here were children doing real programming with a specially designed language and environment. As with Simula's leading to OOP, this encounter finally hit me with what the destiny of personal computing really was going to be. Not

Figure 2: Children programming in logo. Papert [1981]

> a personal dynamic vehicle, as in Engelbart's metaphor opposed to the IBM "railroads", but something much more profound: a personal dynamic medium. With a vehicle one could wait until high school and give "drivers ed", but if it was a medium, it had to extend into the world of childhood. Kay [1993]

3 Collaborative and Augmentative Systems

Users nowadays use several metaphors for networked computers, such as links, bookmarks, hypertext, web, publishing, collaborative editing and so forth. These concepts seem "natural", but they are as artificial as the tools that embody them. We can trace their creation to their modern creator, Vannevar Bush. [2] Bush, in his highly influential 1945 article Bush [1945] elaborated about the memex, which had a stark similarity with the modern personal computer and the internet:

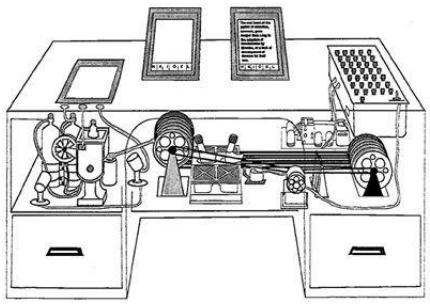

Figure 3: Memex, as it was illustrated for LIFE Magazine, 1945. The desk contains two main microfilm projectors and mechanical apparatus to retrieve the pages for a given trail. Müller-Prove [2002]

> The owner of the memex, let us say, is interested in the origin and properties of the bow and arrow. Specifically he is studying why the short Turkish bow was apparently superior to the English long bow in the skirmishes of the Crusades. He has dozens of possibly pertinent books and articles in his memex. First he runs through an encyclopedia, finds an interesting but sketchy article, leaves it projected. Next, in a history, he finds another pertinent item, and ties the two together. Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. When it becomes evident that the elastic properties of available materials had a great deal to do with the bow, he branches off on a side trail which takes him

[2] Bush as responsible for a great deal more, such as being the man responsible for the creation of the modern american military-scientific establishment. Zachary [1997]

> through textbooks on elasticity and tables of physical constants. He inserts a page of longhand analysis of his own. Thus he builds a trail of his interest through the maze of materials available to him.

But if Bush was the original thinker, the original builder was Douglas Engelbart. His efforts resulted in *Not just hypertext, but graphics, multiple panes, efficient navigation and command input, interactive collaborative work, etc. An entire conceptual world and world view* Kay [1993] Engelbart [1962].

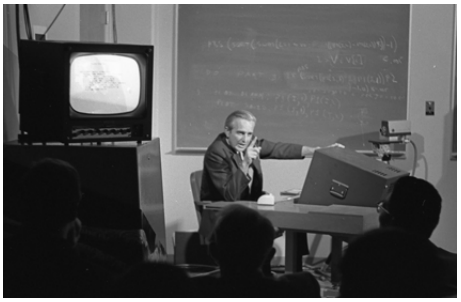

Figure 4: Engelbart demo, 1968.

Engelbart's 1968 demonstration took place in the American Federation of Information Processing Societies' Fall Joint Computer Conference. It shook the world then, and it when on the have an enormous amount of influence in the community that created the personal computer revolution. Müller-Prove [2002]

> The impact of this vision was to produce in the minds of those who were "eager to be augmented" a compelling metaphor of what interactive computing should be like, and I immediately adopted many of the ideas for the FLEX machine. Kay [1993]

4 Conclusion

A sense of history, even a superficial one, is crucial in any endeavor, specially so in an endeavor that requires creative thinking such as programming. As the field looks like it is in constant change and development the practitioner may have difficulty in judging fads from advancements, the new from the rehashing of the old. In doing so he will be less productive and more prone to repeat the mistakes and not enjoy the advancements of the past.

While the material presented here presents some of the past in the area of programming and personal computing, it is merely a glimpse. The programming and technological community needs more focus and studies on its short history for it to progress with more diligence and its members be able to make more informed and wise decisions.

References

Vannevar Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, 1945. URL http://www.theatlantic.com/magazine/print/1969/12/as-we-may-think/3881/.

D. C. Engelbart. Augmenting human intellect: A conceptual framework. Technical report, Stanford Research Institute, October 1962. URL http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html.

Alan C. Kay. The early history of Smalltalk. *SIGPLAN Not.*, 28(3):69–95, March 1993. ISSN 0362-1340. doi: 10.1145/155360.155364. URL http://dx.doi.org/10.1145/155360.155364.

M. Müller-Prove. *Vision and Reality of Hypertext and Graphical User Interfaces*. PhD thesis, University of Hamburg, February 2002.

Seymour Papert. Constructionism vs. instructionism. 1980. URL http://www.papert.org/articles/const_inst/const_inst1.html.

Seymour Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, January 1981. ISBN 0465046274.

Seymour Papert and Idit Harel. *Situating Constructionism*. Ablex Publishing, 1991. URL http://www.papert.org/articles/SituatingConstructionism.html.

Ivan Edward Sutherland. Sketchpad: A man-machine graphical communication system. 2003. URL http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-574.pdf.

G. Pascal Zachary. The godfather. *Wired*, November 1997. URL http://www.wired.com/wired/archive/5.11/es_bush_pr.html.