Whiteboards that Compute: Goals and Challenges for System Designers

Ryan Dixon and Timothy Sherwood
{rsd,sherwood}@cs.ucsb.edu
University of California, Santa Barbara

Even after more than thirty years of desktop computing, many (if not most) engineers facing complex problems instinctively run right to the whiteboard. A whiteboard offers an incredibly natural way by which ideas can be remembered: it provides a means of describing and communicating the relationships between complex interacting parts, and can be used to specify rough schematics of final designs. While the utility of these simple smooth white slabs is clear, there is an opportunity here to create a computing device that can provide all of these abilities and more--a computer that is as simple as sketching but as powerful as a modern desktop. Written equations can be captured and solved, sketches of state machines can be executed, and previous drawings can be queried, recalled, shared, and transformed. We are not the first to claim this is an important direction of work, so why do we believe this to be a fruitful direction for systems research now?

Hardware

From laptops to cell phones, mobile devices have been pushing the limits of small-scale technology. So much so, in fact, that it might be easy to overlook a separate and very different trend that is occurring: the rapid increase in the size and affordability of large displays. These displays, which are constructed lithographically and carved from large pieces of "mother glass" are doubling in area every 1.5 years (See: Figure 1). As the price of "human-scale" displays drop, coupled with affordable pen tracking hardware already available on the market, it will become economically viable to replace a traditional whiteboard with a digital screen.

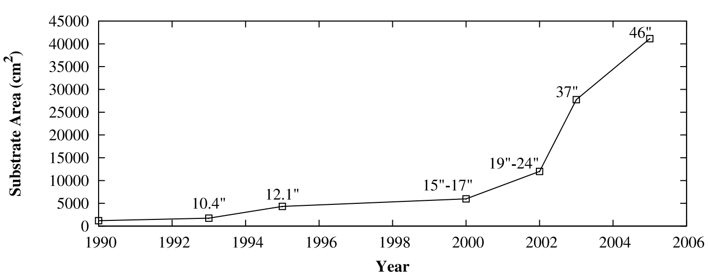

Figure 1: Flat panel display growth rates outlined by AKT [1].

Software

Large, cheap displays are in fact only part of the challenge. To successfully replace a whiteboard, the display must have an incredibly unobtrusive user interface--always ready for simple sketching, capable of making inferences about sketches, and able to act upon these inferences in real time. Over the last 10 years, hand built prototype systems have been created for a variety of domain specific problems, and this first generation of prototypes is just now reaching a product-ready level of maturity. The similarities in these projects will most likely result in a distillation of frameworks, platforms, and shared infrastructure. Architecture, systems, and languages could all play a large part in directing these new developments.

So what specifically is there for our fields to work on in this new domain?

More so than in existing computing paradigms, users will not tolerate fine-grained commands or complicated interfaces. Instead, the burden of disambiguating a user's intentions must be left to the machine [3]. Systems must be designed with imprecision as a first class citizen as the errors inherent to free-form sketch input are enormous. The best-known algorithms for parsing free-form drawings rely on a tight hypothesize-model-measure loop [2] that places an extraordinarily heavy demand on the system. Each reasonable hypothesis must be tested in isolation so that the most likely interpretation of the input can be discovered. New error-conscious parallel primitives are required to make these steps operational in real time. Novel tradeoffs that play between system perceptiveness, human participation, and computational efficiency may be possible in this new domain.

Following this line of reasoning, the interface itself presents some interesting system research opportunities. Managing all of the figures and drawings on the board in both time and space is a very tricky problem. A quick look at a typical white board from our laboratory shows that it is covered with everything from lists and announcements to precise diagrams, sketches used during conversations, even amusing doodles. Appropriately grouping these sketches, moving them around the board as new and old sketches become active, and searching through prior sketches are effectively open problems to different extents in the HCI community. Each stroke generated by the user will likely contain tens to hundreds of points alone. With display resolutions reaching into the millions of pixels, a board full of drawings can result in a significant amount of data. The storage, query, and manipulation of these sketches (both the actual points and their associated meanings) quickly becomes difficult to reason about and manage. A scalable shared software architecture that can unify these different one-off projects is badly required.

One of the biggest problems facing this line of research is the jump from a series of hand-built application specific techniques to a general purpose framework that supports the creation and extension of novel graphical domain specific schemes. The context in which a sketch is created is a very large part of understanding how the sketch is to be viewed, yet all sketches are constructed in fundamentally the same way. Through inference or explicit user action, techniques for sketching across domain barriers must be developed. Programming languages have the potential to play a double role in this space. Not only will new tools be required to construct digital whiteboard systems, a whiteboard environment could facilitate completely new approaches to visual language design. It remains unclear what such a whiteboard language would look like, or even if one extensible enough to allow the creation of new whiteboard tools, yet practical enough for everyday use, is even possible.

Ultimately, we desire a system that can compute upon free-form input from any number of domains without sacrificing the freedom provided by a conventional whiteboard. We believe this presents a number of new challenges to architect, system, and language designers alike and we're wild and crazy enough to believe we can make it work.

References

[1] AKT. (2005). AKT Large Area PECVD Capability, Retrieved July 3, 2007 from http://www.appliedmaterials.com

[2] Christine Alvarado and Randall Davis. Dynamically Constructed Bayes Nets for Multi-Domain Sketch Understanding. *IJCAI-05*, pages 1407-1412, 2005.

[3] Scott Elrod, et al. Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations and Remote Collaboration. *CHI'92*, pages 599-607, 1992.