Style and Function of Graphic Tools

Ted Selker

The Media Laboratory
Massachusetts Institute of Technology

Abstract

Shouldn't the future be a place where our graphical interfaces disencumber computer use? We should be developing a library of graphical presentation and interface techniques relative to where they are useful. We should work to make things respect the ergonomic and psychophysical realities of people. We should work to make things that look like what they do or represent. When these goals over constrain design, we need good teaching tools - prosthetics - to help the user share the designer's vision.

When should the user interface should slink out of the way to allow us to focus our attention on our tasks and when should it be stylish and playful? The sea-green institutional paint of the 50s was supposed be a relaxing color. Interfaces also might suffer from being offensively bland. We choose and use things to make a social statement of status, style, and knowledge. This paper discusses how tools should give designers the latitude to create brand and style statements, while making reasoned and motivated choices of interface techniques in scenarios.

Key words: 3D interface, graphical interface, spatial metaphor, icon, visual language.

1 Introduction

Traditionally, design is an *apprenticeship craft* taught by mentors. Mentors know how to create things and pass that knowledge on through shared practice. Where papers and theories are the legacy of psychophysics, artifacts tend to be the legacy of fields such as architecture, graphic design, or industrial design. The quality of the designs and an ability to teach design can be compelling. The ability to record how to accomplish design when the master is not there can be elusive.

Graphical communication on computers and computer interface design have developed in this apprenticeship way. Many say that the success of the Apple Macintosh interface had to do with exemplars, a tool kit, and a team committed to following guidelines to make application user interfaces work like each other. The value of a consistent and repeated style and language is undeniable [22]. Historical consistency can save relearning

Historically, designers make interfaces that imitate each other and are based on the tools to which they have access. Design often follows exemplars to create a vernacular style, like the Macintosh and Windows GUI's of today. This tendency to have a vernacular style is inevitable and useful. However, it can also keep design from incorporating new possibilities. For example, the selection-speed advantage and aid to memory that pie-sector menus offer [19] over pull-down menus has been ignored by the mainstream for more than a decade. If interface graphic design is not accompanied by interaction design, then interfaces have beautiful icons that aren't integrated into a coherent graphical language. The design moment could be a useful time for the system to teach designers about tradeoffs. How can we make tools that incorporate best-of-breed explanations of how and why to use what tool in which situation?

The Human Computer Interaction (HCI) field is slowly identifying ergonomic facts, explanations of how some functionality might be best exposed, statements about how to make "help" truly useful and ways in which style can match the social goals of the user. Design education can and should move from a human mentoring approach of style families to a field based on scientific underpinnings. Good design requires understanding where tools and techniques will not work as well as where they will. The graphics and interface fields are beginning to reach out to the psychophysics community to achieve this.

Since a position paper for building a design-oriented literature for visual interfaces cannot be comprehensive. I refer to a few articles as examples that show the kind work that can aid in creating a scientific craft to replace the apprenticeship craft in graphical interface engineering.

2 Visual Language

Visual language is the systematic use of visual presentation techniques to convey meaning. In the 1980s we ran an exploration called Visual Representation and Epistemology of Presentation at the IBM T.J. Watson Research Center. The thought was that understanding the mapping between representation and presentation would improve visual interfaces. The work began with

defining terms: What is a visual interface? What makes one different from another? What is visual communication? How can it be systematically analyzed? Our definition of visual language followed.

Definitions of visual language have taken many forms. Some have taken a formal mathematical approach to defining visual language [2], [14]. MacKinlay used first-order logic to define a formalism for working with presentation graphics [26]. While task action grammars have been useful for evaluating interactive systems [9, 41, 84], such formal definitions of visual grammars do not necessarily afford a working definition or feeling for visual language. Chang segmented visual languages relative to what they do [10]: languages that support visual interaction, visual programming languages, visual information processing languages, and iconic visual information processing languages. Selker, et. al., [38] built a framework for comparing systems with visual interfaces relative to the visual elements used in an interface, expressive power of the interface, interaction style, visual interface's relationship to other visual interfaces, relationship between the interface and system functionality and domain of application.

Visual elements, the graphical linguistic parts of a graphical interface, describe its complexity and consistency. Bertin's work [4] described a vocabulary for the elements of diagram, network, and mapping [39]. Visual Elements Workspace (VIEW) [40] is a system for building and exploring a visual language in graphical interfaces. Visual elements consist of the graphical alphabet, interaction alphabet, graphical syntax, interaction syntax, and structure of the language. The visual elements are a language that is constructed as part of graphical interface design.

Graphical alphabet is the set of spatial, surface, and temporal techniques used in a visual language. Iconic systems [10] and visual objects define similar distinctions. Techniques for producing visual alphabets [15, 25, 4, 39] include: imagery and surface characteristics.
Imagery can be representational or abstract. Lakin's 'visual parser' allows shapes to be interpreted as meaningful language elements in a visual grammar [24]. *Images* are derived from photos, digitization or projections. *Icons* are symbols with physical world referents. Icons are stylized representations of objects or processes. For example, the fork and knife icon denotes a place to eat. *Symbols* are drawings of arbitrary design [25]. The character "a" is an example of a symbol. These are labels with no a priori referent

Surface characteristics of color and rendering are the presentation characteristics that has consumed large resources in the computer and computer field.

Color has the perceptually salient components of hue, saturation, and value. Color components are often used to convey information, for example red is associated with stop. *Texture* on computer interfaces, like texture on wallpaper, can be composed of images, icons, or symbols. Halftoning, for example, is a process that uses dichromatic texture to recreate gray-scale images. A grid of varying size dots gives the illusion of a gray-scale image. Texture can render or differentiate things on a screen. *Rendering* is the technology and approach used to present a visual language. The field of computer graphics has focused on issues in rendering. Techniques for rendering include drawing with vectors or polygons. Layout languages such as OpenGL [34] are languages for rendering. Hooper [18] demonstrates differences in various presentation techniques for similar imagery.

Interaction is how a visual languages alphabetic primitives are placed and moved in a visual utterance. The input language, application (semantics) and visual language (output) are what a user interacts with.
Visual languages can be input only (invisible), output only (depictive), interactive (manipulable), or generative. Discussions of interaction have ranged from interactive vs. batch [32] to the use of direct manipulation [42]. 'Direct manipulation' is a closed-feedback loop with a transparent mapping between action and consequence, and temporally indistinguishable response time. Rouse discusses issues of appropriate feedback and response in some detail [37].

Input syntax - is part of a scenario for presenting change in a visual language. Input languages are dependent on and defined by the way they interact with a person. Input syntax is the sequence of actions for communication from a user to the system. The visual output techniques described above need to be matched by some set of input techniques. These techniques include selection, gesture and shape interpretation.

Keyboard Entry includes using keys to enter symbols to control the interface. The keyboard entry device may be any device with buttons, e.g. keyboard or keypad, which maps to an input language.

Point and Pick is the use of menus to control an interface. Most graphical interfaces are simply spatial menus. Selection of menus are made with a mouse, joystick, digitizing tablet, touch screen or other direct manipulation technique. Menus allow a user to select

items from a list, labeled buttons, a set of pictures, or structured spatial display [43].

Point and Move is selecting and moving a graphical object. Many windowing systems allow the user to point to an object and drag it to a new location on the display. This technique was first demonstrated in the Sketchpad system [45]. Many systems (video games, CAD systems, etc.) use gesture, shape, and temporal movement to dictate the "parse" of an image.

Point and draw: is the general drawing technique, which leaves a mark wherever the cursor has been. Grail was probably the first system to demonstrate the use of drawing in an interface [12].

Gesture interpretation is the use of motion through space and time. Typical mouse drivers use the speed with which the mouse is moved to accelerate cursor movement on the screen. Handwriting recognition systems can use gesture recognition to interpret characters and editing commands [46, 48, 49]. Handwriting can also be analyzed through static image interpretation techniques.

Feedback and response time can entirely change the semantics of a visual interaction. The language is different if a person moves a pointing device and it presses back, or if the pointing device doesn't even move the cursor for 2 minutes.

Interactivity abstraction A relatively direct correspondence between action and response exists when a person is using a screen-based editor. The computer echoes the symbols the person types and moves the cursor around the screen when the arrow keys are pressed. An indirect correspondence exists in a text editor when a person types on a command line and the computer responds with a message elsewhere on the screen.

Visual syntax - is the combination of visual techniques, or alphabet, into visual statements. Linguistic terms are used in this paper solely as points of reference. Natural language relies on articles, prepositions, and punctuation for delineating structure and context. Visual language, on the other hand, is parsed relative to position, size and time.

The spatial relationships between visual elements confer information to the observer include:

Positional relative relationships are used in the familiar the icon and menu interfaces: *Sequential* – Spatial sequence of objects denote an order in which visual elements should be considered. Written languages use sequential syntax. Words follow each other left to right and top to bottom. Sequential languages can use space differently. Cartoons frames are often arranged on a line or column. *Metrical* - Measurement is used to denote relative value. Plotting a function uses metrical syntax. Plots using polar, Cartesian, projected 3D and logarithmic scales are examples of metric coordinate spaces. A metrical syntax can use position on a B-spline as its metric. Graphs and bar charts use a Cartesian coordinate system to relate two dimensions of information *Orientation* - Angular relationships between visual objects can drive a visual parse as well. For 3D solids, the rotation presents a view of objects or a scene. A rotation of 180 degrees along the horizontal or vertical axis would result in a view of the far side (rear) of the object. For an analog clock, the orientation of the hands with respect to the dial is the indication of the time of day.

Positional interacting relationships include the way a cursor changes when it is inside a window, or the windows affect each other when they overlap: *Embedded* - Spatial enclosure or geometric containment is often used to define a visual statement. Examples include text balloons in cartoons. VennLisp is a programming language using an embedded notation [24]. A program is defined by positioning hollow language key symbols inside of each other. Evaluation in VennLisp proceeds from the outside of the diagram inward. *Intersecting*: - Partial or total spatial overlap of visual objects is another popular way of determining what a visual statement means. In electrical circuit diagrams, the intersection of straight lines at right angles indicates continuity in the circuit. Venn diagrams use intersecting regions, as well as embedded syntax, to express relationships. Overlapping window systems often use intersection (occlusion) syntax to define which windows are active.

Positional denoted specifiers, such as lines or labels, indicate relationships between objects. *Connected* - Arcs can describe connectivity between visual objects. These connections specify semantic information. Node/arc notations are used in computer flow charts, electrical wiring diagrams, organization charts, etc. *Labeled* - Spatial relationships can be maintained by symbols. In large circuit diagrams, spatial relationships are not always seen on a single drawing. Labeled wires are used to refer to other drawings.

Size – The scale of objects influences how and when component items are interpreted (parsing order). Catalogs often contain large pictures of expensive objects, smaller ones for less expensive. Ancient petroglyph languages sometimes use size to show relative importance of people and things on cave walls.

Shape – Spatial relationships, like those found in puzzles, allow syntax to prescribe how things can be connected together.

Rule –. Visible and invisible algorithms describe visual relationship. For example cursors in text editing environments might stop at the end of a line, disappear, wrap around to the next line or to the top of the screen. The visual meaning of each chess piece is spatial but defined by special rules.

Temporal – Many visual languages use time as an organizing dimension. Computer drawing programs use temporal information to affect meaning in several ways. Recentness can be used to determine the order of spatial overlapping. Blinking draws the attention of the reader. Most mouse oriented scenarios present some functionality through temporal actions, such as pressing the mouse button twice to start an application.

Structure of a system of visual languages

Most visual interfaces are composed of heterogeneous interacting visual languages. Menus are often augmented with text entry or other direct manipulation techniques. Deciding how languages are used for different purposes in a visual interface is part of visual language design. Use of separate visual languages in a system can segment function, system structure, type of manipulation or perspective on information.

Coverage of a visual language ranges from special to general purpose [44]. A paint system uses the same visual interface to draw anything. A special notation for describing organic chemical bonds might have less coverage but give a system more expressive power where it is used. A keyboard calculator has a one to one mapping from space to input syntax. The Instruments subsystem written for LOOPS. broke up its interface syntax using a structured icon shows all of the functionality of the system. By choosing various parts of this structured icon, a user accesses various subsets of its capabilities.

An instrument is chosen to view and change a variable in a program. When an instrument, such as the pitcher (pitcher fullness is the indication of a variable's value), is chosen, the coverage of the instrument is one variable, not the whole program.

Visual syntaxes can be combined in relationships using rules or constraints. JUNO [33]. And ThingLab [6] use "constraint" languages to define the spatial relationships between displayed objects. Things can 'attract' nearby objects. Many systems define two one way relationships rather than symmetric constraints [32].

The VIsual Elements Workspace (VIEW) [39] was created as an interactive system for creating interfaces based on the elements of visual language. The goal was to separate choosing interaction techniques, visual alphabet and visual syntax. The last decade has gone from no graphical interfaces to having an online world of imagery and interfaces in a tense competition for attracting attention. Selker's framework [39] did not consider utility and social appropriateness which have become even more important this decade.

The rest of this paper revisits issues of visual language with respect to utility and acceptance. Future tools for creating visual interface need to help address ergonomics issues as well as visual language issues.

3 Ergonomics

Even though the most basic personal computer of today renders animated 3 dimensional (3D) interfaces most of today's user interfaces are 2 dimensional (2D). How do we gain the skills and confidence to use the new powerful techniques as they are produced? Several areas have been studied. Color, time, spatial organization, and metaphor demonstrate ergonomic things we should respect when designing interface.

A thesis of our work is that tools of the future will respect our perceptual and cognitive abilities. Work is being compiled that will bring psychophysics to the designers. Bernice Rogowitz [36] and her team compiled perceptual interactions of color hue and saturation to create a graphical intelligent tool for choosing color scheme. The system and the user work together to choose a color scheme. This mixed-initiative scenario allows the user to control which perceptually reasonable color scheme to use for their Graphical User Interface (GUI). Enns' perceptual work [13] shows that the dropped shadows we put on icons improve acquisition in a super-fast precognitive way. He also shows several other precognitive effects that are not yet used in GUIs .

4 Spatial Qualities

Location can be an important aid for the recall of information. Mandler et al. [28] and Hasher and Zacks [16] found that people automatically encode spatial information. Since forming spatial associations is a pre-attentive process, designers should use this to their advantage when creating interfaces.

Experienced users do not seem to read menu item names. As novices users primarily rely on the names of the menu items to accomplish simple tasks. Initially, users tend to focus on local attributes like icon shape, but over time switch to identifying attributes that are global or require the user to consider the interface as a whole like the icon's relative position to other icons on the screen [31, 21].

Ark et al. [2] show that with experience, people can find selectable things faster in a three dimensional image than a 2D field of icons. The work shows that

making selections on a 2D regular array of icons was slower than on a 3D spatially organized depiction. Variation in pattern of placement, variation in shape of selectable item, cognitive landmarks were also found to make improvements in selection speed [3]. The value of shape in 3D arrangement and landmarks relate to memory because these differences are developed with experience.

Ark et. al. [3] also worked out that the effects of the non-regular layout landmark and 3D realistic representation are additive. The subjects' performance was better when *any* of these techniques were added. Interactions between techniques are important for design. Work should show the contexts in which results are or are not valid. Better understanding of the value and interactions of techniques to usability improves the possibilities of creating design aids. Rogowitz's color selector mentioned above is a rare example of embedding such usability findings in a tool [36].

Graphical techniques can improve usability making things more discernable or memorable. Remarkably, the often quoted Miller's [30] classic 7 plus or minus 2 paper actually describes short term memory being enlarged in 3D. Maybe people would have been using 3D as a memory aid earlier had 3D not the number 7 been in the title of George Miller's landmark paper.

5 Visualization

It is useful to line up columns to do arithmetic on paper or on a spread sheet. It is common to use lights arranged on Cartesian coordinate to make sense of the volume of low and high frequencies on stereo systems. Popular programs include many uses other uses graphical visualization techniques from the relative color of the frame of an active window to the size and position of the slider in a scroll bar.

Standard visualizations such as pie and bar charts are now a mouse click away in many computer tools. A performance analysis tool we call the Memory Hierarchy Framework [1] demonstrates the value of spatial mappings. This approach shows constraints of data movement though a computer architecture spatially. It uses a concise drawing composed of rectangles and connecting segments to do this. Traditional modeling of data transfer through computers models transfer time as a constant from memory for performance analysis. The framework shows the packing constraints that cause the memory movement time to vary. Disk, main memory, memory internal to the processor, etc are represented in modules. Each module represents its capabilities by a vertical size, a horizontal count, and a leg connecting to the next module with length for how long it takes to transfer data. By using logarithmic space, one image can show an observer issues of memory transfer over many orders of magnitude in memory capability. The consequences of analyzing architectures and algorithms with this tool are dramatic. Seeing how data gets partitioned as it moves through the hierarchy in a famous computer architecture called into question why one memory layer, the Transition Look aside Buffer (TLB), had a different aspect ratio than other layers. The engineering team revisited and changed the architecture to make subsequent products run algorithms better. The visualization was also used to show how to reduce memory thrashing to realize more than an order of magnitude speedup in a commercial compiler for matrix multiply, and several other algorithms.

6 Prosthetics

Good visualization technique, realistic graphical rendering, and metaphors can all improve interface effectiveness. When the graphical interface is not understood, help and tutoring prosthetics are needed. Associating explanatory popup text bubbles with graphical objects has become popular.

Annotation is a particularly interesting approach because it uses an auxiliary visual language to improve the primary one. We might expect that overlaying two interfaces would be confusing. The technique of adding demonstration animations has been used extensively. Slug trails [41] have been an evocative approach for teaching graphical techniques. A diagrammatic on-screen caricature of a mouse blinks its eyes to represent user button actions. It leaves its image on the screen whenever it changes which buttons are pressed. This mouse guide moves, leaving an annotated dotted arc of 'slug trail' to describe its action. This annotating tutor shows the order of actions in a graphical procedure. An on-screen annotation of surrounding icons with small so-called icon dressing animations was an idea to reduce the obtrusiveness of tutoring graphics. A four pixel wide bezel would surround the icon. The embellished icon would then animate in the bezel to describe change that could happen to the object. Rather than putting the image elsewhere, the goal was to allow the user to continue focusing on the thing they were working with. This idea turned into a technique of overlay used in OS/2 SmartGuides [41].

The usefulness of semitransparent overlay was experimented with. Covering parts of a window made searching for all information in its dialog box fields faster. Searches through the overlays with up to 80 percent density were not slowed. Searches for things which had no overlay on them were as efficient as if they were

the only thing in the window [51]. A secondary annotating visual language can be very effective.

7 Interaction

People need to stay oriented while focusing on a task. Input action depictions, their feedback, and their mapping to function are critical to user orientation.
Adding feedback to button presses of the TrackPoint for example improves speed of acquisition by a significant amount [8].

Mapping position from an input device to a physical or graphical space can be productive. If the space is too large, a positional mapping gets clumsy. Consider replacing the gas pedal with a squirrel-cage-like barrel. The driver would have to continuously roll the barrel with their feet to travel. This rolling to move an indeterminate distance would require a lot more effort than pressing the rate-controlled gas pedal. In the case of scrolling to move through a long document or web page on a screen, the situation is the same; scrolling with a rate-controlled joystick rather than a wheel improves graphical acquisition time by more than 20% [51]. Graphical interface mappings matter.

How does a user navigate in 3D? A joystick and a trackball used to sit on the graphics lab tables. The gain and self-centering features would be tuned for different applications. Current 3D interfaces tend to presuppose the value of untuned 6 degree of freedom joysticks (6DOF) with helicopter like control mappings. Requiring this practiced skill every time viewpoint is changed might be unnecessary. The unconstrained (positional orientation) input language and the (ecological 3D) output language contribute to make constrained activity difficult. Traveling down halls and around a standard environment becomes perilous.

An attempt to improve the 3D navigation problem with metaphor and constraints at the input device was shown using joysticks with a bulldozer metaphor. Pulling forward and backwards on both joysticks moves and turns user orientation in a plane. Pushing in and out moves the orientation outside the plane. Using two 2D joysticks and the bulldozer like control approach improved 3D navigation over using a standard 6DOF control Zhai et al. [52]. 3D input can be improved with mapping and metaphor.

A visual prosthetic for 3D interfaces

The original Core graphics standards supported multiple views. Intermediate views can be used to keep users oriented in 3D graphical interaction. In the above visual language terminology this might be called an intermediate positional orienting depictive only language with an associated constrained movement input language

Assume a set of images including a tower that can be looked down from, a road that presents a 3D path to a 3D location (needing no visible supports) and jumping spots that can be transported to. These simple mechanisms and improved and orienting experience could be transferred to many 3D control environments. The input language would allow the input to stay in the 2-D world of walking and jumping around that people are used to.

Consider a world with moons The user is stand on the ground. To look up, the user tilts to move the view off the surface. A few moons and planets are in the sky with barely visible magic roads reaching up towards them. A user chooses a path with the cursor. Pressing forward moves towards the moon. When the moon has been reached a landing pad spot can be left, as a spot to jump to. Standing on the moon the user selects a tower to get a view of the surrounding area. A pad in the center of a crater is useful for jumping to. This scenario uses visual reminders and constrained motion to simplify 3D motion. A simplified interface can improve exploration of a space.

8 Style: a memory & learning aid

A lot of our work is aimed at making things not violate psychological and physiological constraints of people. One unspoken theory of the HCI field theory has been that ergonomics will solve all problems. NOT; choosing color schemes only for psychophysical properties (e.g. contrast, legibility and resolution) alone promotes a simple, functional world. Fortunately people want more, interfaces that make them want to do things.

A calendar drawing representing The Wizard of OZ - has a *mustard yellow sky*! The artist knows that skies are not yellow. Possibly they chose yellow to refer to the yellow brick road that is in the story; possibly they chose yellow to be dramatic. Color is used to encode information, to help make things stand out, and as decoration. People might take a long time choosing background color on a slide. Why not? They took a long time choosing which color clothes to wear today too. Just selecting the background effects to appear on a slide for a talk takes us longer than we would like to admit. Memory and learning specialists tell us to associate things we want to learn or remember with something funny or odd [23].

As graphics and user interface experts, we need to explicitly embrace oddness and style as one of the most important mechanisms for associating two things. Style can attract and keep peoples' attention by helping make things memorable.

Onomatopoeia maps the sound made saying like "shushing" to what it represents. Such mappings can improve the interpretability and learnability of words. The word's sound reminds one of the word's meaning. The memory map is special. Direct manipulation spatially map input motion to their effects. Such a mapping reminds the user of the meaning and can simplify user experience. To date, most work on creating techniques to map directly between user actions and computer actions have been contained successes. Rarely are new interfaces so technically wonderful that they fits well into a current computer use culture, be it that of programmers, designers, professional or end users. Since culture changes slowly, why should it be surprising when it takes time for new ideas to become widely used. Sometimes we feel stifled and work to define a new culture that expands what we feel able to do. Sometimes we create an actual cultural change that people want but will take a few generations to accept.

The last several decades have created computers that can represent space, surfaces, and things in time; output devices that create realistic still and moving representations of the real and manufactured world and input devices and languages to manipulate these worlds. All these things are improving, but the excitement is in discovering how to integrate these things into the vernacular.

We are on the cusp of exploring how graphical interface will give us insight into our universe. We are starting to learn how to use it to explore fantasies. We are extending the way to use these techniques in order to make work more efficient and useful. For 3D interfaces we need not push against an old culture, we get to define it. People want better ways of making interactions, visualizations, presentations, etc. Still it shouldn't be surprising when some visualization scene that has gotten traction slows acceptance of a technically superior one. Historical consistency is one of the most important reasons for keeping or adopting an interface approach.

9 Conclusion

The future feel of graphics tools must be that fluid, effortless feeling of competence. It rests on the substrate of fabulous rendering and presentation technology. It rejoices in the power and variety of techniques for presenting and interacting graphically. It will be powered with libraries of presentation, visualization and interaction techniques. It will be shaped by understanding where these techniques make ergonomic and functional sense. But techniques and approaches don't solve problems by themselves.

The future feel of graphical interface will follow needs. Progress begins with the efficiency of simplified processes and procedures. It moves easily where graphics interfaces can replace and extend older or non-computerized approaches. New approaches also will be adopted for the glamour of seeing new things and new ways to understand them. New approaches, however, are still accepted by fits and starts of novelty rubbing against the status quo.

People will use interfaces they are accustomed to. Simplifying interfaces, visualization, imagery and metaphors are balanced against the confusion of their unusualness. The only relief for the pressure of the existing is the fashionableness of the new. We can make fashionable and fad inspired things. It is still early in the graphics and user interface fields; our fads can be the quality designs that they will call the status quo.

We are hopeful that:

1. Good user interface is taking the tool out of tasks.
2. Making things look like what they are or do can help them be useful.
3. Matching the ergonomics and psychophysics of humans improves interfaces.
4. Prosthetics that help, tutor, and coach can be built to remember what a user has learned and to teach people when visualization and interface are more sophisticated than intuitive.

Such a perceptual match approach would be blind without taking into account social/political reality. The way we use computers is and will be part of our statement about ourselves. Who will augment their reality when talking to others? The tools we use are a personal statement. They tell others what we care about, know, and want. Style does matter.

References

[1] Alpern, B., L. Carter, E. Feig, and T. Selker. The Uniform Memory Hierarchy Model of Computation. *Algorithmica*, vol. 12, pages 72-109, 1994.

[2] Ark, W., Dryer, D. C., Selker, T., Zhai, Shaman. Representation Matters: The Effect of 3D Objects and a Spatial metaphor in a Graphical User Interface. In *People and Computers XIII, Proc. of HCI'98*, H. Johnson, N. Lawrence, C. Roast (Eds.), pages 209-219, 1998a.

[3] Ark, W., Dryer, D. C., Selker, T., Zhai, S. Landmarks to Aid Navigation in a Graphical User Interface. In *Proceedings of Workshop on Personalized and Social navigation in Information Space.* 1998b.

[4] Bertin, J. *Semiology of Graphics: Diagrams, Networks and Maps*, 1983.

[5] Bolt, R.A. Spatial Data Management System. Final Technical Report, U.S. Defense Advanced Research Projects Agency, 1978.

[6] Borning, A. Thinglab: A Constraint-Oriented Simulation Laboratory. Ph.D. Thesis, Department of Education, Stanford University, 1979.
[7] Butler, T. W. Computer response time and user performance. In *CHI '83 Proceedings*, pages 58-67, 1983.
[8] Campbell, C., S.. Zhai, K. May, and K. Maglio. What you feel must be what you see: adding tactile feedback to the TrackPoint. *INTERACT'99.*
[9] Card, S. K., T. P., Moran and A. Newell. *Psychology of Human Computer Interaction*, 1983.
[10] Chang, S. K. Visual Languages: A Tutorial and Survey, *IEEE Software*, pages 29-39, 1987.
[11] Cole, I. Human Aspects of Office Filing: Implications for the Electronic Office. In *Proceedings of the Human Factors Society - 26th Annual Meeting*, 1982.
[12] Ellis, T. O., and W. L. Sibley. On the Development of Equitable Graphic I/O. *In IEEE Transactions on Human Factors in Electronics*, pages 15-17, 1967.
[13] Enns, J. T. and R. A. Rensinck. Scene-based properties influence visual search. *Science* 247, pages 721-723, 1990.
[14] Fu, K.S. Languages for Visual Information Description. In *1984 IEEE Computer Society Workshop on Visual Languages*, pages 222—231, 1984.
[15] Gittins, D. Icon-Based Human-Computer Interaction. *International Journal Man-Machine Studies, pages* 519—543, 1986.
[16] Hasher, L., and Zacks, R.T. Automatic and effortful processes in memory. *Journal of Experimental Psychology: General*, 108, pages 356-388, 1979.
[17] Hess, S. M., Detweiler, M. C., and Ellis, R. D. The Effects of Display Layout on Monitoring and Updating System States. In *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting*, pages 1336-1340, 1984.
[18] Hooper, K. Experimental Mapping The Perceptual Representation of Environments. Technical Report, University of California, Santa Cruz, 1982.
[19] Hopkins, D., J. Callahan, and M. Weiser. Pies: Implementation, Evaluation, and Application of Circular Menus. University of Maryland Tech Report, 1987.
[20] Kaehler, C. MacPaint Manual. Apple Computer, Inc., 1983.
[21] Kaptelinin, V. Item Recognition in Menu Selection: The Effect of Practice. In *INTERCHI '93 Adjunct Proceedings*, pages 183-184, 1993.
[22] Kellogg, W. A., and T. J. Breen. Evaluating user and system models: Applying scaling techniques to problems in human-computer interaction. In *Proceedings of CHI+GI 1987: Human factors in computing systems and graphics interface*, pages 303-308, 1987.
[23] Klatzky, R. *Memory and Awareness: an information processing perspective.* 1984.
[24] Lakin, F. Visual Grammars for Visual Languages. In *AAAI '87 Proceedings, Vol. 2*, pages 683—688, 1987.
[25] Lodding, K. N. Iconics- A Visual Man-Machine Interface. In *Proceedings of the National Computer Graphics Association*, pages 221—233, 1982.
[26] MacKinlay, Jock D. Automatic Design of Graphical Presentations. Ph.D. Thesis, Stanford University, 1986.
[27] Malone, T. W. How Do People Organize Their Desks? Implications for the Design of Office Information Systems. In *ACM Transactions on Office Information Systems*, vol. 1, no. 1, pages 99-112, 1983.
[28] Mandler, J. M., Seegmiller, D., and Day, J. On the coding of spatial information. *Memory & Cognition*, 5, pages 10-16, 1977.
[29] Marcus, A. *Tutorial 18: User Interface Screen Design and Color.* ACM/SIGCHI, 1986.
[30] Miller, George A. The Magical Number Seven Plus or Minus Two: Some limits on our capacity for processing information. *Psychological Review*, vol. 63, pages 81-87, 1956.
[31] Moyes, J. Putting Icons in Context: The influence of Contextual Information on the Usability of Icons. Ph.D. Thesis, University of Glasgow, 1995.
[32] Myers, B. A. Creating Dynamic Interaction Techniques by Demonstration. In *CHI '87 Proceedings*, pages 271—284, 1987.
[33] Nelson, G. Juno, a Constraint-Based Graphics System. In *Proceedings of the ACM/SIGGRAPH Conference*, 1985.
[34] Palevich, J. JavaOpenGL 1.0a3. ©Copyright 1997 .
[35] Reisner, P. Human Factors Studies of Database Query Languages: A Survey and Assessment. *Computing Surveys*, 13, 1983.
[36] Rogowitz, B. E. and D. A. Rabenhorst. CRAFT: A tool for customizing color and font selections guided by perceptual rules. In *Proceedings of the IS&T and SID Color Imaging Conference*, 1993.
[37] Rouse, W. B., Human-Computer Interaction in the Control of Dynamic Systems. *Computing Surveys*, vol. 13, no. 1, 1981.
[38] Selker, T., C. Wolf, and L. Koved. A Framework for Comparing Systems with Visual Interfaces. *Interact '87 Proceedings*,1987.
[39] Selker, T. Visual Elements Workspace (VIEW) Scenario. IBM Tech Report RC15513, 1990.
[40] Selker, T., and A. Appel. Graphics as visual language. In *Handbook of Statistics*, vol. 9, 1993.
[41] Selker, T., R. J. Barber, and R. J. Kelley. Effective, selective presentation of help material in a graphical environment: experince with COACH/2, a graphical adaptive help system. IBM Tech Report, 1996.

[42] Shneiderman, B. *Software Psychology*. 1980.
[43] Shneiderman, B. *Designing the User Interface*, 1986.
[44] Shu, N. C. Visual Programming Languages: A Dimensional Analysis. In *Proceedings of the International Symposium on New Directions in Computing, Trondheim, Norway*, 1985.
[45] Sutherland, I. E. Sketchpad: A Man-Machine Graphical Communication System. In *Spring Joint Computer Conference Proceedings*, vol. 23, 1963.
[46] Tappert, C. C., J. M. Kurtzberg, and P. S. Levy. Elastic Matching for Handwritten Symbol Recognition. IBM T. J. Watson Research Center Tech Report RC9988, 1983.
[47] Van Der Veer, G. C. Individual differences and the user interface. *Ergonomics*, 32, 11, pages 1431- 1449, 1989.
[48] Ward, J., and B. Blesser. Interactive Recognition of Handprinted Characters for Computer Input. *IEEE Computer Graphics and Applications*, vol. 5, no. 7, pages 24-37, 1985.
[49] Wolf, C. G., and J. R. Rhyne. A Taxonomic Approach To Understanding Direct Manipulation. IBM T. J. Watson Research Center Tech Report RC 13104, 1987.
[50] Woo, M, J. Neider, T. Davis. OpenGL Architecture Review Board, P. Womack, *Opengl Programming Guide : The Official Guide to Learning Opengl*, Version 1.1, 2nd ed., Addison-Wesley Pub Co; 1997.
[51] Zhai, S., Wright, J., Selker, T., Kelin, S. Graphical means of directing users' attention in the visual interface. In *Proceedings of INTERACT: the sixth IFIP conference on Human Computer Interaction*, 1997.
[52] Zhai, S., B. A. Smith, and T. Selker. Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks. In *Proceedings of INTERACT: the sixth IFIP conference on Human Computer Interaction*,1999.
[53] Zhai, S., E. Kandgon, B. Smith, and T. Selker. In Search of the "Magic Carpet", Design and Experimentation of a 3D Navigation. To appear in Journal of Visual Languages and Computing. To appear in Journal of Visual Languages and Computing.