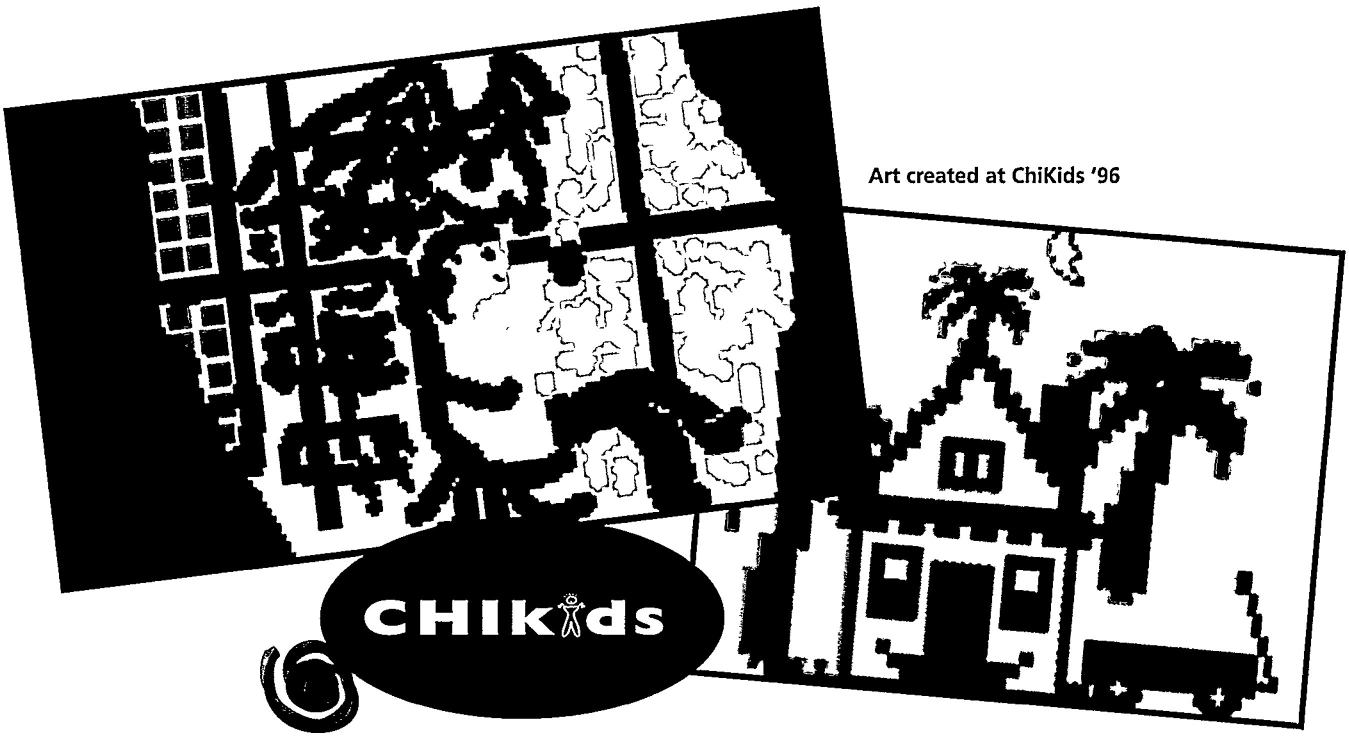
Karen G. Schneider

Children and Information Visualization Technologies



Art created at ChiKids '96

It went zip when it moved,
And pop when it stopped,
And whirr when it stood still;
I never knew just what it was,
And I guess I never will!
—*Tom Paxton*

Information Visualization for Children: The User in Context

Considering the interior lives of children, from a design perspective it may be helpful to think about children as a series of cognitive communities, divided roughly along developmental stages. Piaget's four developmental stages are worth iterating, if only to phrase them in terms of children and computers, with the caveat that children are highly individualistic [13]. These stages are best appreciated, as Anderson says, "as simply ways of characterizing what is inherently a gradual and continuous process" [1]. The first, the sensory-motor stage is less interesting to us; it would be a stretch to discuss human-computer interaction at this developmental stage, since what is "real" to these new humans is affection, physical contact and food.

Preoperational stage: 2-7
The early Nintendo years

Children's mental processes, says Anderson, "are intuitive and lack systematicity" [1]. They cannot reason, they can only hold one item in working memory at a time, their attention span is brief, and their conservation skills are still hampered by an inability to understand abstract ranges, such as an alphabetical row, as well as a predilection to "be distracted by find discrete movement easier than continuous—that is, children were more adept at moving cursors that automatically clicked onto items ("hopping") than they were at sliding a cursor to an item ("walking") [16]. Strommen postulates that "walking" the cursor places additional demands on working memory by requiring children to hold both monitoring and stopping schema in working memory, whereas discrete movement, which requires only monitoring schema as the cursor stops itself, "removes the need to dynamically exchange the monitoring and stopping schema during cursor placement" [16].

Strommen's work highlights the importance of direct manipulation, a useful device for preoperational children for several reasons. First, younger preoperational children are preliterate, so bypassing keyboard input is necessary (though this does not preclude using text on the interface for educational purposes). Second, orthography, overall, does not improve until fifth grade (11 years old) [6], suggesting there may be a cognitive component to spelling skills. Third, enabling the child to work directly with the objects on the interface, rather than indirectly through keyboard commands, reduces the additional information required to be held in working memory.

Concrete-operational stage 7-11 years

Anderson has pointed out that "most neural development is complete by [age] 2 and almost all by age five. Humans are kept children by the slowness of their physical development" [1]. In the concrete-operational stage, we see children maturing to the brink of adult cognitive abilities. Though they cannot formulate hypothesis, and though abstract concepts such as ranges of numbers are often still difficult, they are able to group like items [9] and categorize [2, 3]; among other things, this is when they become expert at browsing activities, which, Borgman notes, are hugely appealing to children (perhaps, I wonder, because abstract searching is still developmentally distant from their skills, or is it because browsing is a natural outcome of the information-hungry brain?). Working memory quick-

the irrelevant features of a display" [1]. They also, toward the middle of this stage, become highly imaginative (4-year-olds are known for their imaginary friends), superstitious ("step on a crack, break your mother's back") rules-oriented ("it goes THIS way, Mom!"), egocentric—which Kuhlthau translates to "individualistic and self-centered" [9]—and delight in things that are obviously incongruous.

Toward the end of the preoperational stage, children begin understanding organizational schemes; a seven-year-old, for example, can understand how books in a library are organized [9], and field reports posted to DYNIX-L, a discussion group for the Dynix library automation system, suggest that older preoperational children are able to use the simple organizational scheme of Kid's Catalog, a graphical interface for library media (though formal studies in this area are called for).

This is also the stage when children begin to develop rudimentary principles of conservation, which, Anderson points out, "appears earlier on some tasks than others. For instance, conservation of number usually appears before conservation of liquid," suggesting that interface design for these children cannot rely on the changed size of objects to convey new knowledge [1]. Finally, preoperational children have limited experience and domain knowledge, key components of perceptual literacy [8]; without this visual fluency, complex visual environments can be cacaphonous and hence confusing. Computer interfaces and input devices for children in the preoperational stage often reflect the limitations of the preoperational child by using only three or four screen colors and objects and making the interface as explicit—or in K. Norman's terms, apparent [11]—as possible. Edmonds et al. [6] demonstrated ably that abstractions sharply reduce interface efficiency for children below the formal operational stage. For the youngest children in this group, Strommen's work with three-year-olds and Nintendo games [15, 16] provides additional and intriguing insights into interface design requirements.

Strommen also found that three-year-olds

Karen G. Schneider is a government contractor (Garcia Consulting, Inc.) and director of the Region 2 Library, United States Protection Agency schneider.karen @epamail.epa.gov

An earlier version of this paper was submitted for partial fulfillment of a grade in Information Visualization Technologies, School of Information, University of Michigan, Fall 1995

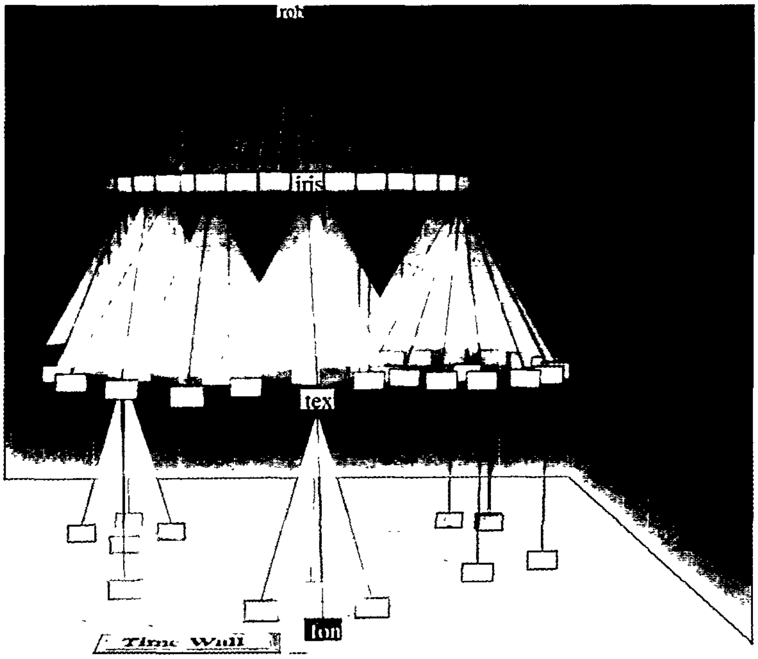


Figure 1
Cone Tree visualization of a directory hierarchy

ly develops, allowing children to manipulate more than one item in memory simultaneously (an achievement that will disappear with graduate-level academic status).

As Kuhlthau notes, this is also an age when children so disposed begin working in groups [9]. We could speculate that increased working memory and ability to work in groups would enable children in the concrete-operational stage to use groupware (although the literature is, typically, silent on this topic). Though the body will continue to develop, motor skills are good, facilitating many types of input devices.

Perceptual acuity is developing, driven by experience in the world, domain knowledge and developmental maturity. This author has observed children with early and frequent exposure to computers in Paterson, New Jersey's School District 4 expertly navigating cluttered Macintosh "desktops" for work and play, not only using specific applications but understanding the interactivity of tools, such as creating an image with one application that can be dropped into another. This suggests that, though high abstractions are still difficult for children at this age, a metaphor with no childhood antecedent can apparently be learned by children, as long as it is internally consistent.

Toms and Kinnucan [17] have emphasized that metaphors can be problematical, particularly if they are poorly-understood by the user, but as Borgman [3] notes, children's "model of the computer appears to be sufficiently uncluttered that it can be adapted to a variety of interfaces." It may even be that the literalness and rules-driven world of the concrete-operational child lends itself to learning a wide variety of arbitrary rules and behaviors that from adolescence onward would inspire extended enquiry if not rebellion.

Finally, in the concrete-operational stage, a child's "appreciation of conservation reaches new levels of abstraction"; this allows them to appreciate "the idealized conservations that are part of modern science," but, as Barry observed in Peter Pan, also closes doors to a certain fantastic universe privy to the illogical and imaginative world of the very young child [1].

It is a truism in English departments that Dickens' fascinatingly well-imagined children grew into boring adults; perhaps he found the passage into the formal-operational stage an uninteresting literary challenge. Though in this report, we are, like Dickens, primarily concerned with the first three developmental stages, it is worth noting that the computing needs of adolescents have been largely under-appreciated except by computer-game developers.

A Sampler of Information Visualization Software and Strategies

Many information-visualization tools, techniques and theories are developed for adult expert users, by adult expert users. Nevertheless, rummaging through discussions of key strategies reveals that there is much to be borrowed, adapted or rethought for the three cognitive communities of childhood discussed in the previous section.

General Strategies

In the long run, it is easier and more humane to adapt the interface to the person, rather than vice versa. Jock Mackinley's "presentation tool," though designed specifically for "effective graphical presentations," offers the notion of algorithm-driven user-specific displays [10]; it is easy to imagine an interface that used several simple queries to determine user skills at login, or even dynamically recalculated and redesigned the interface based on user errors or other behavior, offering data-rich, complex displays for older children with computer expertise and reducing color sets, shapes and quantities of information for children who are younger or less expert.

A related concept is the Cognitive Coprocessor employed in the Information Visualizer [5]. The Cognitive Co-processor is an "interface manager substrate" that can be "tuned" to adjust for "perceptual processing, immediate response and unit task." While the primary function of the Cognitive Co-processor is to ensure the interface runs smoothly, primarily so the user experiences cognitively-useful response time, a MacKinleyesque treatment might include tweaking these values at login or during use to accomodate perceptual ability.

Animation offers several advantages. For the abstraction-sensitive child, there is the obvious advantage of realism, i.e., a duck moves like a duck. Another is object constancy, a tool for "reducing cognitive load"; as an object on an interface moves or is moved, "animation of that motion makes it possible... for the user to retain the relationships of what is displayed" [14a]. This is particularly crucial for children as it minimizes the demands on limited working memory which would otherwise overflow from the effort of "reassimilating the new display" [14a]. Finally, there is pedagogical value in a display that engages the user in its activity, as long as there is not so much motion, shape and color as to overwhelm the cognitively immature.

Three-dimensional interfaces, used judiciously, are promising. The first response of someone who has worked with children to many 3-D workspaces is that they are abstract, busy and informationally dense, and rely on metaphors foreign to children (will many children, for that matter, grow up to use traditional desks and file cabinets?). However, these problems, on reflection, are trivial to resolve. A stripped-down 3-D interface, reduced to several familiar shapes and colors resembling a child's bedroom or playspace, might offer a realistic space to work in; this is not so much a metaphor as an imitation of life, with games in toychests, homework on a desk, a trash can for disposing of old material, etc.

Another tactic would be designing workspaces like well-known children's books, such as Goodnight Moon, which is detail-rich but very familiar to many children, who enjoy scrutinizing and memorizing the minutia. The key temptation to be avoided would be to use 3-D workspaces for their primary function for adults: to compress more data into a limited area. This Tuftean goal is understandable but is antithetical to the needs of an interface for children, where simplicity and minimal data representation is a goal.

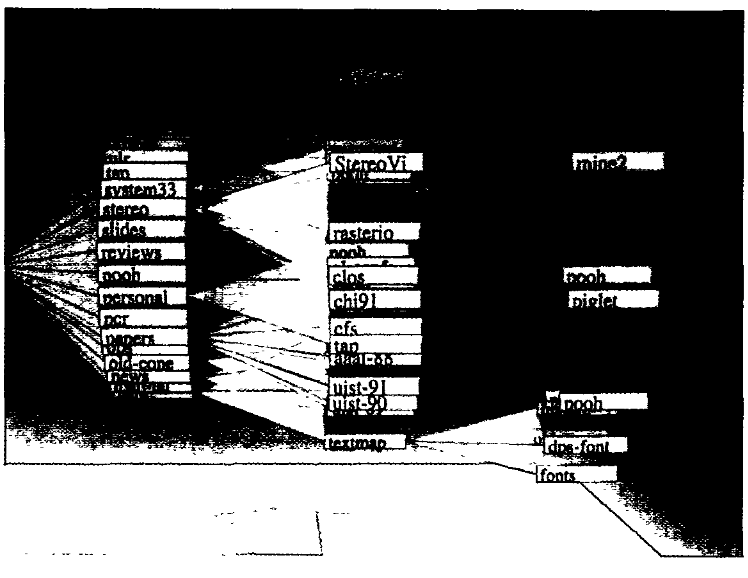
Finally, tools that reflect computational wear—that (to oversimplify) demonstrate when tools have been used and resources have been accessed—have the potential, if used judiciously, to help young users by providing an explicit relationship between actions and outcomes. Hill and Holan (1992), for example, describe Read Wear and Edit Wear, two computational-wear tools. While their examples are fairly abstract, the concept is valid, and with simpler, more apparent display mechanisms would port very easily to the juvenile computing environment. Color, size, shape or popup messages are four simple tools to exploit toward this end; again, the greatest challenge is to provide the concept without miring the interface in toto in too many good ideas.

A Sampler of Information Visualization Tools for Children

Cone Trees

Cone trees [14], three-dimensional animated hierarchies, so clearly lend themselves to interfaces for children that I am surprised not to have seen them so used. They appear to be

Figure 2
Horizontal Cone Tree visualization of a directory



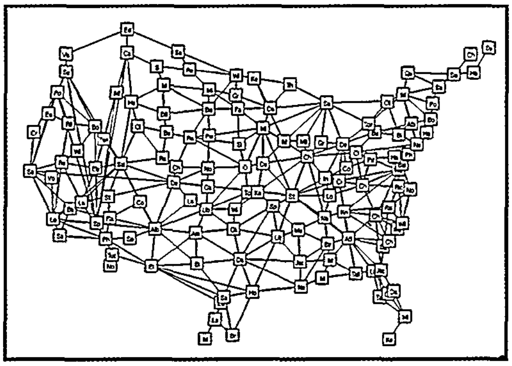


Figure 3
A graph with 134 vertices and 338 edges. The vertices represent major cities in the United States, and the edges represent paths between neighboring cities.

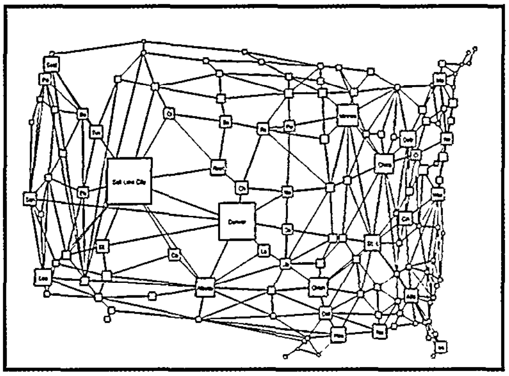


Figure 4
A *fisheye* view of the graph in Figure 3.

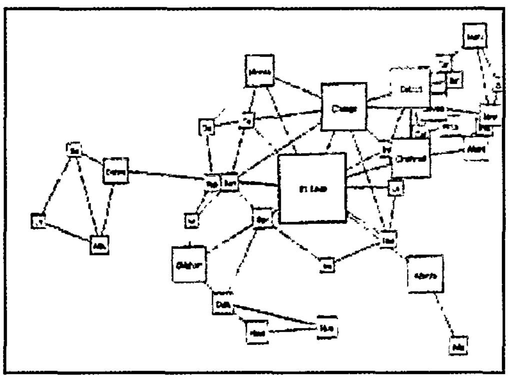


Figure 5
A fisheye view of the graph in Figure 3 with unimportant vertices eliminated.

devices children would naturally reach for and interact with: moveable circular objects festooned with graspable tags. Scaled down with smaller data sets, tailored with iconography from the world of the child, and perhaps augmented with accompanying sound, naturally-playful cone trees could easily become "discovery carousels" (with horses that moved to calliope music and had saddlebags filled with neat things); or perhaps, with such high motivational potential, combined with the child's ability to categorize, cone trees could risk enough abstraction to be more traditional data-storage devices exploiting shape and color to teach children about subject organization. Perhaps if the cone trees moved periodically of their own accord, making their function and use highly apparent, this would counteract the abstraction; or perhaps, instead of shapes, a type of animal could be assigned to each level—"zebras are for homework, monkeys are for my favorite games." As with the "grownup" cone trees, lighting cues, "like lighter coloring of closer nodes and links," and "idealized shadows" 'thrown' by the cone tree would enhance the sense of perspective and versimilitude; this could be extended to objects that "lit up" as they were touched by the pointing device.

Pad

Pad (and its successor, Pad++) is an interesting two-dimensional tool that offers a highly configurable surface and two important tools: portals, which can filter information where they are placed, and zoom action, which allows smooth, variable-speed, animated motion from one object to the next [12]. For children, Pad offers a definable, bounded surface which could be imprinted with outlines, such as geographic areas, calendars and the like. Preconfigured portals and associated tags for Pad objects,with a query engine, could result in this comfortable pattern: the child creates a resource; PAD asks what kind of resource it is; the child tells Pad; Pad tags the object; the child can now retrieve the object via the appropriate portal. There are, shall we say, some minor programming details to work out here, but designing the object-tagging as a series of discrete schema preserves the objective of minimizing simultaneous demands on the immature working memory.

Fisheye Views

Children's egocentric worlds make fisheye views seem naturally appealing. The primary problem is that fisheye strategies gain much of their information from the user, whose understanding of the original representation provides context for the degree of interest (DOI) [7]. The graphical fisheye map representing the United States depicted in Brown and Sarkar [4], for example, would be meaningless to a child whose geographical knowledge did not include a prior concept of this country's shape. For older users, placing an outline of the undistorted representation alongside or transparently underneath the fisheye view remedies the domain-knowledge issue, though it does not address the fundamental abstractness of these representations. A slightly more explicit strategy would be to use three-dimensional and animation techniques, where in response to command input, the object to be "fisheyed" would dynamically distort itself. It is possible, however, that fisheye views lend themselves to interfaces for children only when the representation is highly specific to begin with.

Conclusion

Information visualization strategies for adults, once freed of adult goals, demonstrate high pedagogical and entertainment potential for children. Additionally, studies with tools tailored to children's cognitive needs would, as Strommen's work did, teach us more about how the mind and body work together in new environments. The greatest obstacles to employing such tools are, first, that commercial software for children is designed to be sold to adults, and second, that institutional software is designed by teachers and others who focus on formal learning more than on exploration and discovery.

Bibliography for Children and Information Visualization Technologies

1. Anderson, J.R. Cognitive Psychology and its Implications: Fourth Edition. W. H. Freeman and Company, New York, New York, 1995.

2. Borgman, C. L. Designing an information retrieval interface based on children's categorization of knowledge: a pilot study. In Proceedings of the American Society for Information Science Annual Meeting, 1989, 26: 81-95.

3. Borgman, C. L., Bower, J., Auth, M. J., & Krieger, D. From hands-on science to hands-on information retrieval. In: J.Katzer, G. Newby (Eds.), Proceedings of the 52nd American Society for Information Science Annual Meeting, Vol. 28. October 29-November 2, 1991, Washington, DC. Medford, NJ: Learned Information, 96-103.

4. Brown, M. and Sarkar, M. Graphical fisheye views. Communications of the ACM. 37:12, 1994, 73 -84.

5. Stuart K. Card, George G. Robertson and Jock D. Mackinlay, The Information Visualizer, An Information Workspace, Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems, Information Visualization, 1991, pp. 181-188.

6. Edmonds, L., Moore, P., and Balcom, K. M. The effectiveness of an online catalog. School Library Journal, 36:10, 1990, 28-32.

7. Furnas, George W. Generalized Fisheye Views, Human Factors and Computer Systems, CHI 89 Conference proceedings, 1989, pp. 16-23.

8. Gibson, E. J. Principles of Perceptual Learning and Development. Meredith Company, New York, New York, 1969.

9. Kuhlthau, C. C. Meeting the information needs of children and young adults: basing library media programs on developmental states. Journal of Youth Services in Libraries, 2:1 1988, 511-57.

10. Mackinlay, Jock, Automating the Design of Graphical Presentations of Relational Information, ACM Transactions on Graphics, Vol. 5, No. 2, April 1986, pp. 110-141.

11. Norman, K. Interface Apparency and Manipulatability: Cognitive Gateways through the Spatial Visualization Barrier in Computer-Based Technologies. http://www.lap.umd.edu/LAPFolder/NSFIA/proposal.html, 1995.

12. Perlin & Fox. Pad: An Alternative Approach to the Computer Interface, In Proceedings of SIGGRAPH'93, ACM Press, 1993, 57-72.

13. Piaget, J. and Inhelder, B. Psychology of the Child. Basic Books, New York, New York, 1969.

14. Robertson G. G., Mackinlay J. D., and Card S. K. Cone Trees: Animated 3D Visualizations of Hierarchical Information, in Proceedings of Human Factors in Computing Systems CHI 91, ACM Press, 1991.

14a. Robertson G. G., Card, S. K., and MacKinlay, J. Information Visualization using 3D interactive animation. Communications of the ACM, 36:4, 1993, 56-71.

15. Strommen, E. F., Razavi, S. and Medoff, L. M. This button makes you go up: three-year-olds and the Nintendo computer. Applied Ergonomics, 1992, 23:6, 409-13.

16. Strommen, E. F. Is it easier to hop or walk? Development issues in interface design. Human-computer Interaction, 8:4, 1993, 337-352.

17. Toms, E. G. and Kinnucan, M. T. The effectiveness of the electronic city metaphor for organizing the menus of Free-Nets. (Unpublished paper. ca. 1994)

THE FIGURES IN THIS ARTICLE PREVIOUSLY APPEARED IN *COMMUNICATIONS OF THE ACM.* REPRINTED BY PERMISSION.

Index to Advertisers

Please contact the advertisers directly for more product information

ERRATA for *interactions* III.4

The color study "answers" in last issue's Design Column *(interactions* iii.4) were not complete. The complete answers are as follows:

Blurring: 1 & B
Vibration: 2 & J or 2 & A
Color Relativity: 3 & H or 3 & A

The plate images used as openers for the departments and columns in the last issue (interactions iii.4) did not have copyright information attributed to them. The copyright information is as follows:

PLATE I – c1992 D. R. Fowler, P. Prusinkiewicz, J. Battjes. (Goatsbeard)
PLATE II – c1992 D. R. Fowler, P. Prusinkiewicz, J. Battjes (Table of cacti)
PLATE III – c1994 R. Mech, P. Prusinkiewicz, M. James (Topiary garden at Levens Hall)
PLATE IV – c1992 D. R. Fowler, P. Prusinkiewicz, H. Meinhardt (Volutoconus bednalli, Oliva porphyria, and Conus marmoreus)
PLATE V – c1996 R. Mech and P. Prusinkiewicz (A climbing plant growing on a tree)
PLATE VI – c1996 R. Mech and P. Prusinkiewicz (Horse chestnut without leaves)
PLATE VII – c1992 D. R. Fowler, P. Prusinkiewicz, J. Battjes (Cauliflowers and broccoli)
PLATE VIII – c1994 R. Mech, M. James, P. Prusinkiewicz. (Knot garden at Moseley Old Hall)
PLATE IX – c1996 R. Mech, P. Prusinkiewicz (Horse chestnut with the leaves)
PLATE X – c1992 D. R. Fowler, P. Prusinkiewicz, J. Battjes (Green coneflower)

In the last issue's Book Preview (*interactions* iii.4) the contact information for "Design Guide for Multiplatform Graphical User Interfaces" was omitted. The information is as follows:

Published by Bellcore
$49.95 (plus shipping if outside the U.S.)
Voice 800–521–2673 from the U.S.
and Canada, +1–908–699–5800 from elsewhere.
Fax +1–908–336–2559 or +1–908–336–2692
Bellcore Customer Service
8 Corporate Place, Room 3A–184
Piscataway, NJ 08854–4196 USA