xybrap.html
98.06.25 (d18bz

What's On My Mind

(Invited talk at the first Wearable Computer Conference, Fairfax VA, May 12-13, 1998.)

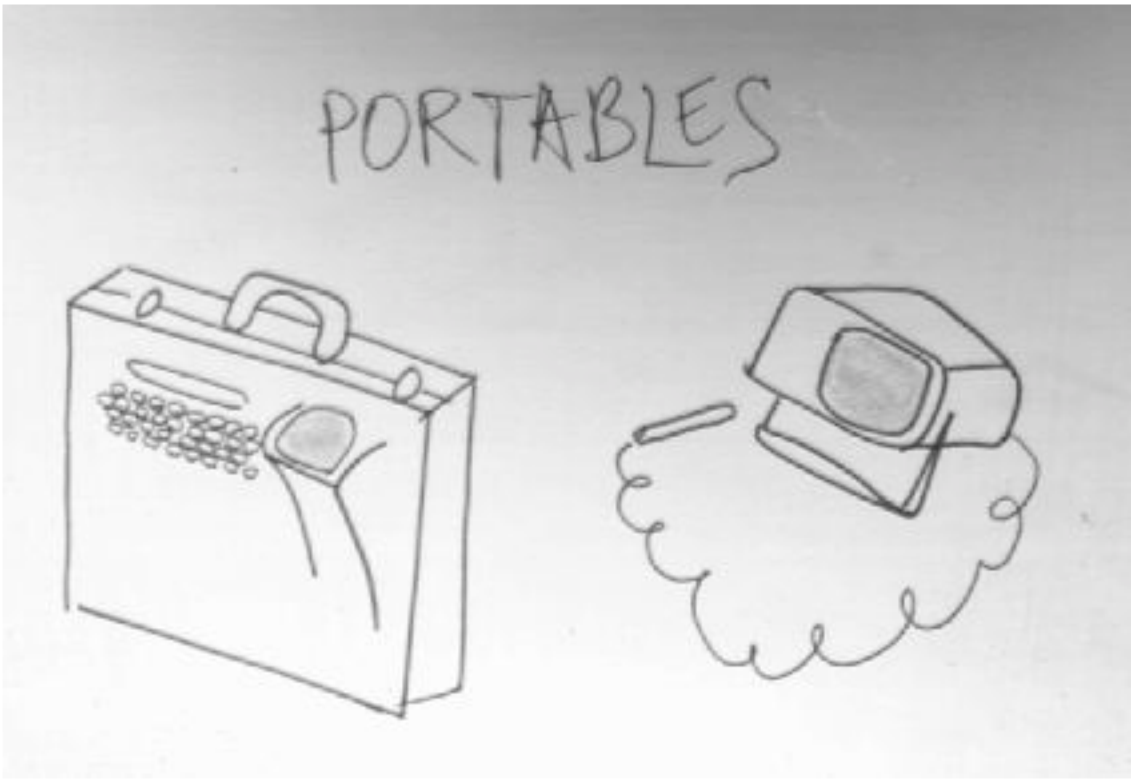
Theodor Holm Nelson
Keio University (Fujisawa, Japan) and Project Xanadu (Sausalito, California)
http://www.sfc.keio.ac.jp/~ted/ and http://www.xanadu.net

I've wanted a Wearable since 1960. Thank you, Xybernaut.

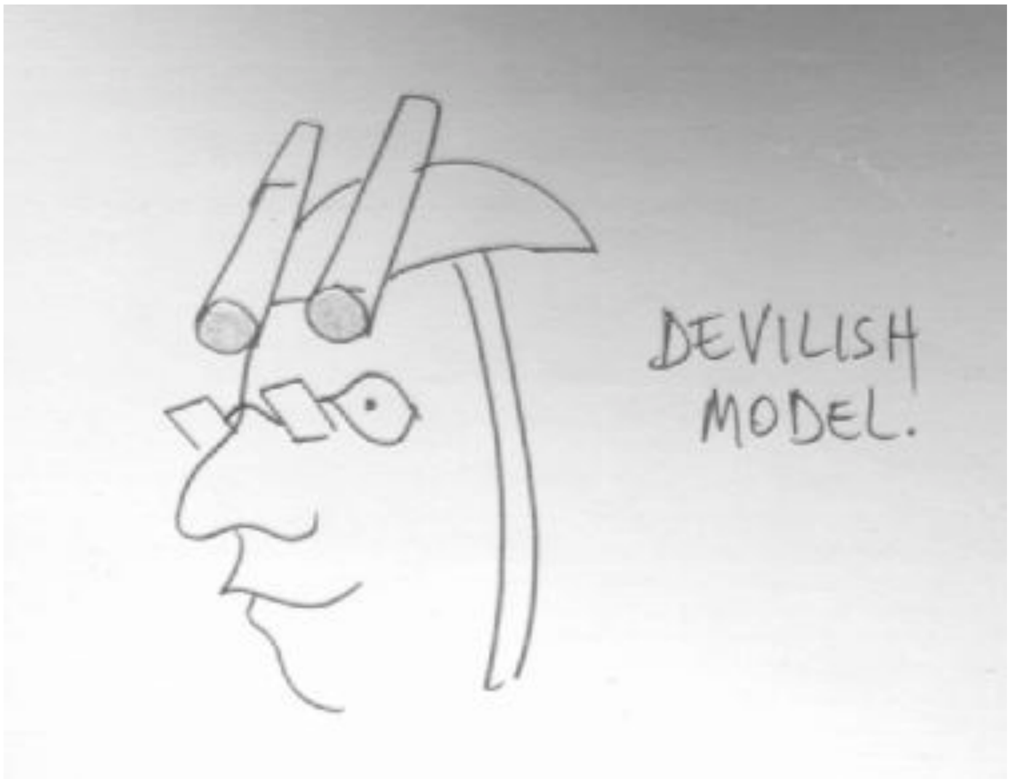
I'm wearing a tuxedo with my Xybernaut for this talk today because that's how I feel about it. Wearable computers should not be just for the factory floor. In the near future, the Wearable computer will be a standard part of the thoughtful person's social ensemble.

To show how my heart has always been in these ideas, I've dug up a couple of slides from 32 years ago.

In 1966, I made the following slide to show my impression of what portable computers could be like.

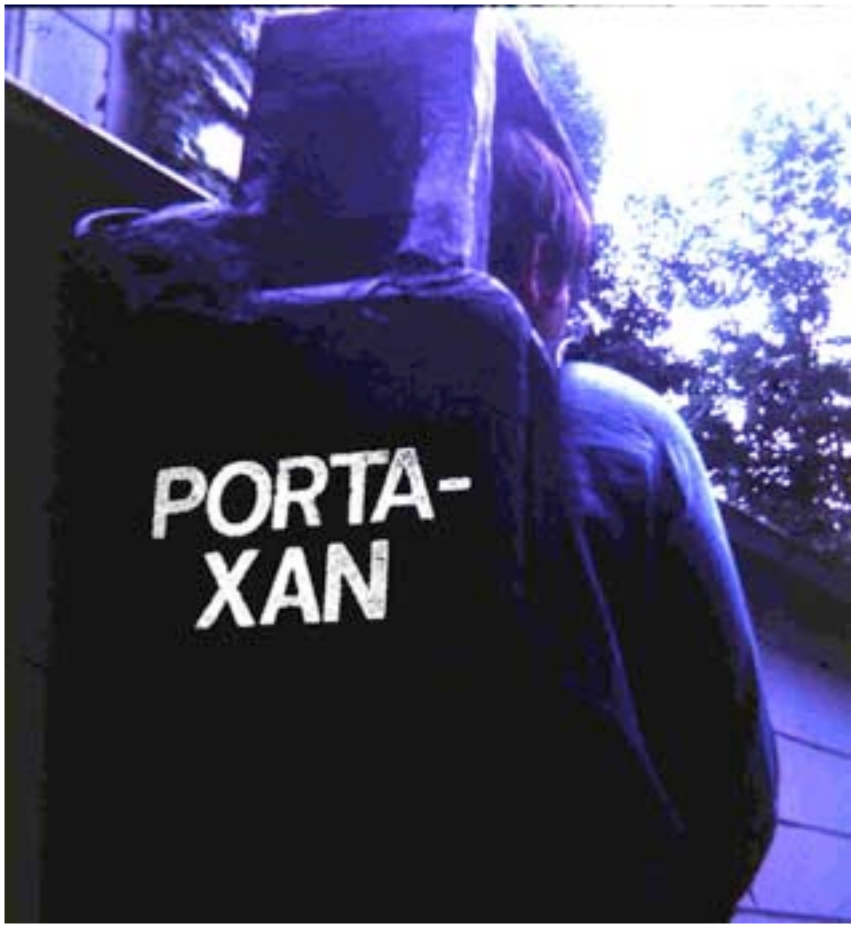


Now for the next slide I showed in that 1966 presentation. Here was my sketch of a wearable computer, with two CRTs and semisilvered mirrors.



(Something structurally very like this, but not as raffish, is on display today in the lobby of this conference-- a sort of pince-nez with two mirrors and two little LCD panels.)

Finally, here's a physical mockup of a portable, wearable computer that a young fellow named Tom Barnard constructed to my design in 1973 or 1974. This picture was in the first edition of my book *Computer Lib*, 1974. The picture shows my wearable mockup, the Porta-Xan, being worn by my brother Daniel.



This wearable computer mockup is large and clumsy, about the size of a briefcase, with hooks over the shoulders and what was meant to look like a CRT (actually in the mockup it was a big flashlight) perched just on Dan's left shoulder. In front of his face (not visible) is a piece of plexiglas, representing a semi-transparent reflector for looking back at the CRT while you also look forward at the world.

But I consider this a perfectly valid design, especially for a computer this size. (And remember that you couldn't even buy a computer this small at the time the model was made, unless it was military or majorly industrial. Only two years later did the Apple II come out, which was about this size. Indeed, this could have been a way of packaging an early Apple for the road.)

Gwen Bell of the Computer Museum has told me that the Porta-Xan may have been the first mockup of a portable computer and that she wanted it for their collection. However, it's kind of shrivelled up now, and perhaps, at 0 megahertz, uninteresting to today's frenzied computer fans.

A RADICAL VIEW OF COMPUTING

But the bigger question is, what's it all for? Why do we use these danged things anyway?

Just because a lot of computers are sold does not mean they make any sense. People accept what they are offered because they cannot imagine better.

I think that the computer field has lost its way, big time. The low-grade, tangled solutions that have been offered for most consumer problems have come to seem god-given. "Word processing" is a geek's notion of what you need for text; the real problem of version management is pushed out into your lap. "Applications" are more intended to trap you than to help you. Computer files are big indivisible lumps in irrelevant hierarchical arrangements, absurdly messy to deal with.

I believe that what most people want and need is a unified system for all their work and records, not a bunch of separate "applications." This means the management of versions and copies and their variations, built of tracked small pieces with their origins known-- not big lumps with their origins lost (files as we know them today). This means juggling and managing vast amounts of fine-grained information-- thousands of items-- in multiple contexts. An item should only need to be entered once, and become available in all its contexts at any time. And all the contexts of an item should be simultaneously visible.

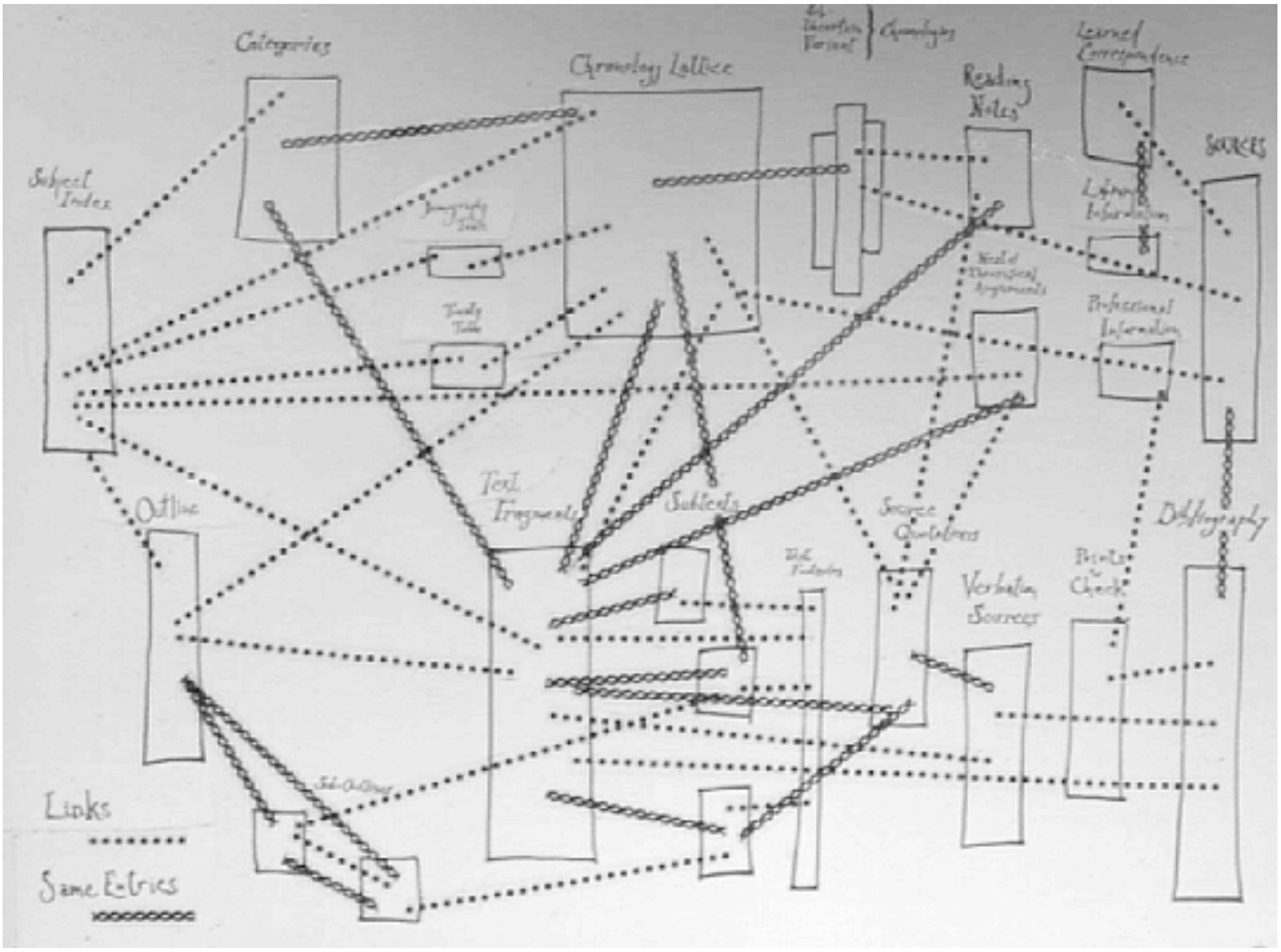
I am not talking about "database" the way it's done today. I'm not talking about ***anything*** the way it's done today. I'm talking about a different kind of system, like the systems I have been designing for thirty-eight years, designs which have gone in very different directions from the rest of the computer world. People have jeered at these ideas, sometimes saying that I didn't understand computers. I contend that in fact it is they, the conventional computer people, who did not understand computers, meaning that they have never understood the way computers would have to connect to our everyday life, and that they have not realized what vastly different software we would need in our lives from the stuff that's out there now.

In my first published paper in 1965 (1), I presented such a design-- what is now called an "information architecture"-- for the organization of personal information. No one understood it. I believe that if people had understood those ideas then, the computer world would not be the mess it is now. But there were so many ideas in that paper-- it also introduced the words "hypertext" and "hypermedia"-- I only begin to see now why they understood so little.

The title of the paper was "A File Structure for the Complex, the Changing and the Indeterminate," and it offered a design for interconnecting all your textual materials-- not in files, but in what I called "zipper lists", an exact data structure. Zipper lists had their items connected sideways; it's an interesting design, but we'll skip the details.

The following illustration-- (this is my original, incidentally, not the one they printed, all squared up to look more "technical")-- was intended to show the complexity of the connected materials that a working personal-computer user would have to deal with. It shows the interconnections that might concern a typical academic user-- say, an historian. (The illustration predicted e-mail, incidentally.)

Each of the rectangles is a zipper list; the funny lines indicate sideways connection bundles. .



The captions are a little hard to read. From left to right, the zipper lists are:

- Subject Index
- Outline
- Categories
- Sub-Outlines
- Demography Table
- Treaty Table
- Text fragments
- Chronology Lattice
- Subtexts
  - (here meaning "portions of text"-- this was before the word "subtext" came in to mean "hidden viewpoint").
- Text footnotes
- Sub/Uncertain/Variant Categories
- Source Quotations
- Verbatin Sources
- Reading Notes
- Nest of Theoretical Arguments
- Points to Check
- Learned Correspondence
  - (Note that this part of the illustration was a prediction of e-mail.)

Literary Information
Professional Information
Bibliography
Sources

(A note in the lower left identifies two different kinds of connection bundles, which we can skip.)

Okay, that was a very long time ago, and I won't tell you the adventures I have had trying to get zipper lists-- and the designs that have come later (2)-- implemented.

But the question is ***still***, how do we handle personal information? Because today's proliferation of specialized computer tools is for most people impossible and therefore useless. There is no answer for those who want simplicity.

Moreover, there is no determinate way to use all the different software tools. Anything can be done in hundreds of different ways, and there are many divergent, vehement, sectarian views about the right ways to use the different kinds of software that exist.

But I happen to believe that what's necessary can't be done by any of today's software at all.

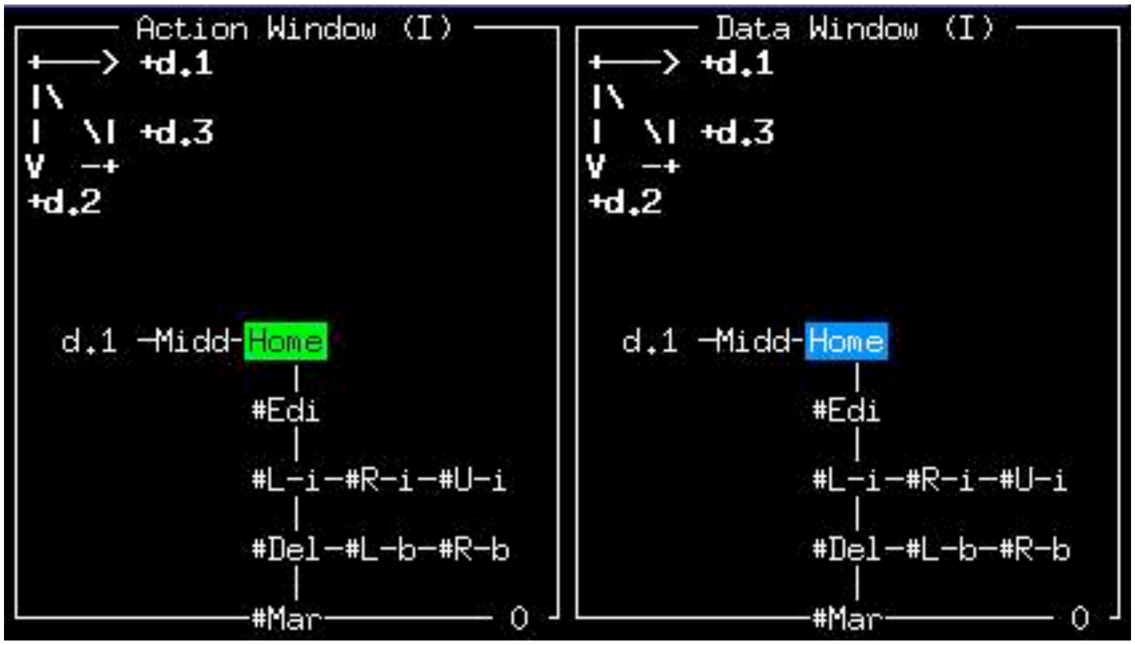
THE ZIGZAG$_{TM}$ SYSTEM

I think there's a case for a simple and unified world, handling many thousands of small items, without application traps. Most people need it. So let me show you a prototype that implements some interesting ideas, intended to lead to such a new kind of simple and unified world, possibly to permit the unification of everything that non-computer people want to do with computers.

This is not an application, but an ***environment***; an environment in which you can do all sorts of things powerfully. But please don't mistake its current appearance for what it will look like later. The current appearance is intended to show the simple basic structure of this environment. Later it can look many different ways, but always with the same consistent structure underneath.

This consistent structure, the ZigZag space-- I'll offer a different name for it later-- may be thought of as a multidimensional generalization of rows and columns, without any shape or structure imposed.

Here's how it looks right now, when you turn it on without opening any previous data--



```
┌────── Action Window (I) ──────┐ ┌────── Data Window (I) ──────┐
+───> +d.1                          +───> +d.1
|\                                  |\
|  \| +d.3                          |  \| +d.3
V  -+                               V  -+
+d.2                                +d.2



  d.1 -Midd-Home                      d.1 -Midd-Home
             |                                   |
            #Edi                                #Edi
             |                                   |
            #L-i-#R-i-#U-i                      #L-i-#R-i-#U-i
             |                                   |
            #Del-#L-b-#R-b                      #Del-#L-b-#R-b
             |                                   |
└──────────#Mar──────────── 0 ┘ └──────────#Mar──────────── 0 ┘
```

These two windows represent different views of the same world: the green cursor on the left is a menu cursor, where you can select an operation you want performed; the blue cursor on the right is the data cursor, for pointing where you want the operation to be done. (The right window is the main window in this implementation.)

But these are just two different views of the same world. The menus are not in principle separate from the data. It's only one world. In principle, too, you may want many views, so the program allows multiple cursors, though at the moment we haven't made windows for them.

The screen dimensions are across (x-axis) and down (y-axis), but you can put any of the ZigZag world's dimensions on either axis of either screen.

Each window can show any two dimensions at a time (and hide a third). In the upper left corner of each window there is a map which shows which dimensions are currently available in the axes of that window.

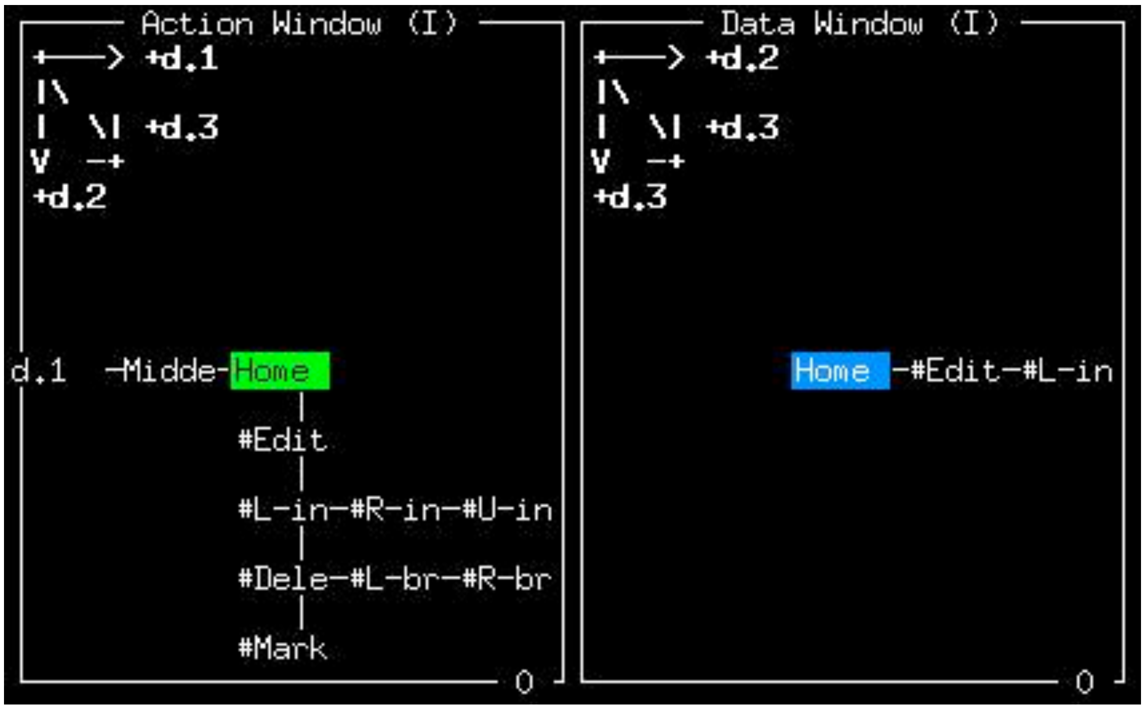
This corner map also shows that in each window you can access a third dimension. In addition to the two dimensions you can ***see*** on the x-axis and y-axis of each window, you have access to a third dimension, forward and back, along which you can move though you can't see it.

But the ZigZag world is not just three-dimensional. It is multidimensional, and you can map any three of these dimensions to a window at a given time.

This is done very simply through the keyboard, using the x and y keys. To change the x-axis of a given window, type "x", and to change the y-axis of the window, type "y".

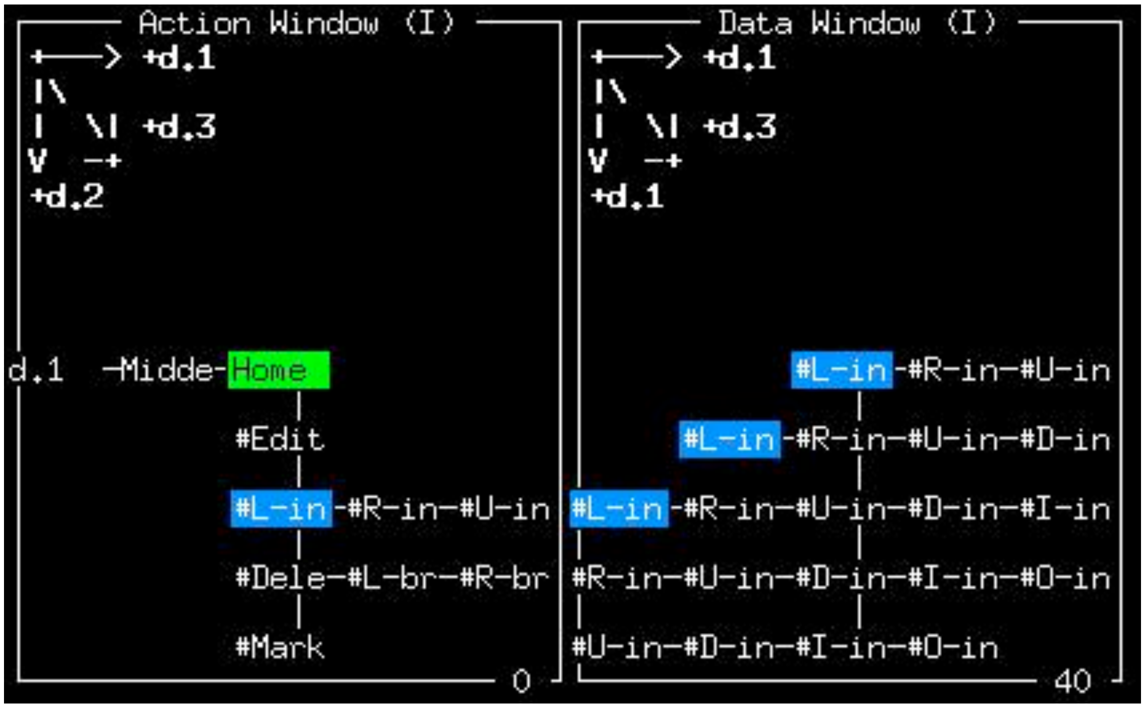
To make this easy in both windows, we currently use lower-case "x" and "y" for the right-hand window, and uppercase "X" and "Y" for the left-hand window. By striking those particular keys, you modify the view of the world in that window. (Why those keys? Because lowercase is easier to type, and the right window is currently the main one.)

For instance, if you start from the view above and hit lower-case "x" and "y" once each, you get this:


```
Action Window (I)                  Data Window (I)
+——> +d.1                          +——> +d.2
|\                                 |\
|  \| +d.3                         |  \| +d.3
V  —+                              V  —+
+d.2                               +d.3



d.1  -Midde-Home                           Home -#Edit-#L-in
            |
            #Edit
            |
            #L-in-#R-in-#U-in
            |
            #Dele-#L-br-#R-br
            |
            #Mark
                              0                              0
```

The righthand view has now changed. What's happened: by hitting the "x" key you've assigned the x axis to the next dimension on the list, which is d.2, and by hitting the "y" key you've assigned the y axis to the next dimension on the list, which is d.3. The result is a different view of the same things that're in the left-hand window: the same cells that go downward from the Home cell in the left window are seen going sideways from the Home cell in the right window. In both cases, the cells are shown along dimension 2 in the positive direction from the Home cell.

What happens if you look at the same dimension on both the x-axis and y-axis? You see an amusing diagonal picture (below, right). It's not very useful, but it's consistent.


```
Action Window (I)                  Data Window (I)
+——> +d.1                          +——> +d.1
|\                                 |\
|  \| +d.3                         |  \| +d.3
V  —+                              V  —+
+d.2                               +d.1



d.1  -Midde-Home                           #L-in-#R-in-#U-in
            |                                   |
            #Edit                       #L-in-#R-in-#U-in-#D-in
            |                                   |
            #L-in-#R-in-#U-in      #L-in-#R-in-#U-in-#D-in-#I-in
            |                      |            |
            #Dele-#L-br-#R-br      #R-in-#U-in-#D-in-#I-in-#O-in
            |                      |            |
            #Mark                  #U-in-#D-in-#I-in-#O-in
                              0                             40
```

Note that the blue cursor of the righthand window (the Data cursor of the Data Window) happens to be on a cell which is also seen in the left-hand window (the Action Window). Therefore the blue cursor shows up in both windows.

MULTIDIMENSIONALITY

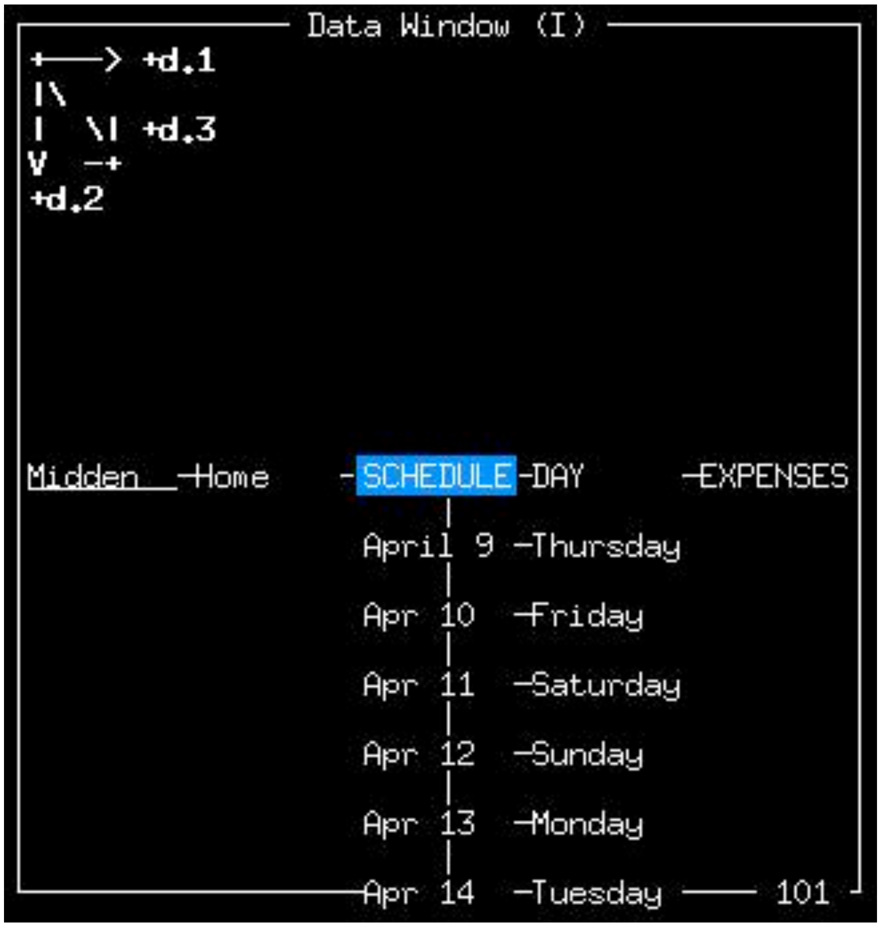
As mentioned earlier, the ZigZag world can have as many dimensions as you like. (You add a dimension simply by putting its name into the dimension list-- as a new cell, of course.) Multiple dimensions may be hard to imagine at first, but if you think about the fact that any two dimensions can present an arrangement of rows and columns you'll begin to get it. Especially if you start playing with the system.

A PRACTICAL EXAMPLE

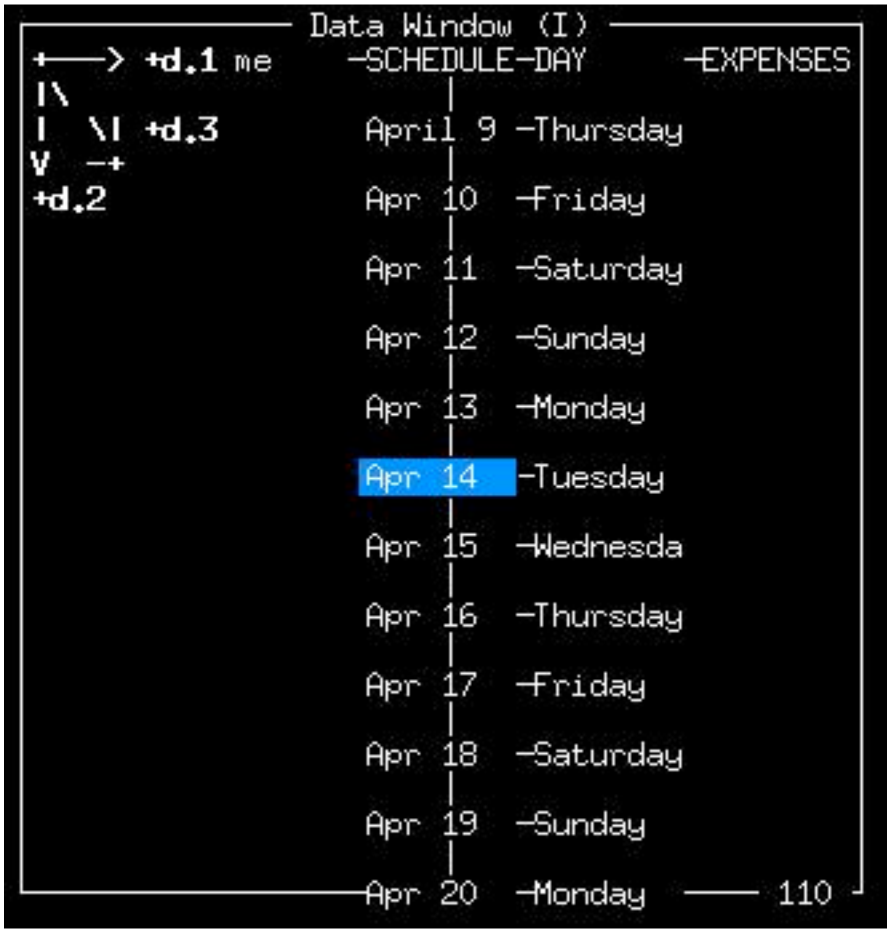
Even in this prototype, the ZigZag world can already be customized for a lot of different purposes. Here's how some new dimensions can make personal record-keeping easier.

Let me show you a nice example, designed by Marlene Mallicoat. We've made a column in Dimension 2 called SCHEDULE (We've also made some others, which we'll get to.)

And we've typed in some cells to show the days of the month and their days of the week.

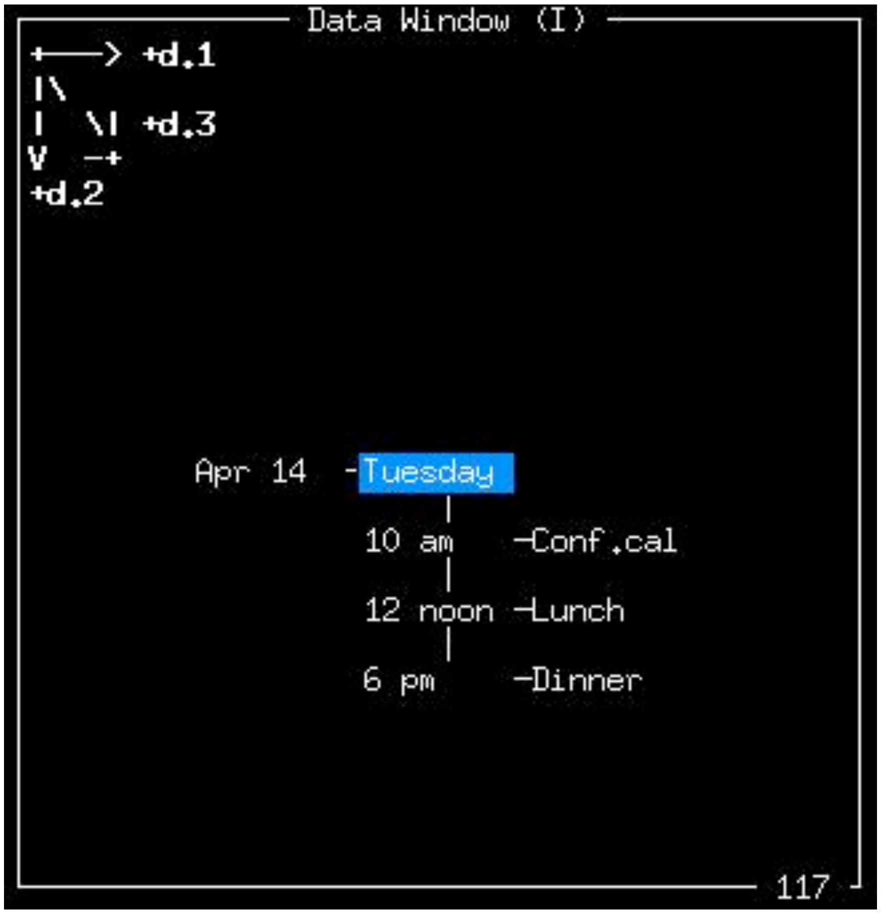

```
                 Data Window (I)
+——> +d.1
|\
|  \| +d.3
V  -+
+d.2






Midden  -Home    -SCHEDULE -DAY      -EXPENSES
                    |
                 April 9 -Thursday
                    |
                 Apr 10  -Friday
                    |
                 Apr 11  -Saturday
                    |
                 Apr 12  -Sunday
                    |
                 Apr 13  -Monday
                    |
                 Apr 14  -Tuesday —— 101
```

As we move the cursor down the column of the individual days, it appears to be an ordinary table.



```
┌──────────────── Data Window (I) ───────────────┐
│+───> +d.1 me    ─SCHEDULE─DAY      ─EXPENSES│
│|\                   |                         │
│|  \| +d.3        April 9 ─Thursday            │
│V  ─+                 |                         │
│+d.2              Apr 10  ─Friday              │
│                      |                         │
│                  Apr 11  ─Saturday            │
│                      |                         │
│                  Apr 12  ─Sunday              │
│                      |                         │
│                  Apr 13  ─Monday              │
│                      |                         │
│                  Apr 14  ─Tuesday             │
│                      |                         │
│                  Apr 15  ─Wednesda            │
│                      |                         │
│                  Apr 16  ─Thursday            │
│                      |                         │
│                  Apr 17  ─Friday              │
│                      |                         │
│                  Apr 18  ─Saturday            │
│                      |                         │
│                  Apr 19  ─Sunday              │
│                      |                         │
└─────────────────Apr 20  ─Monday  ──── 110 ┘
```

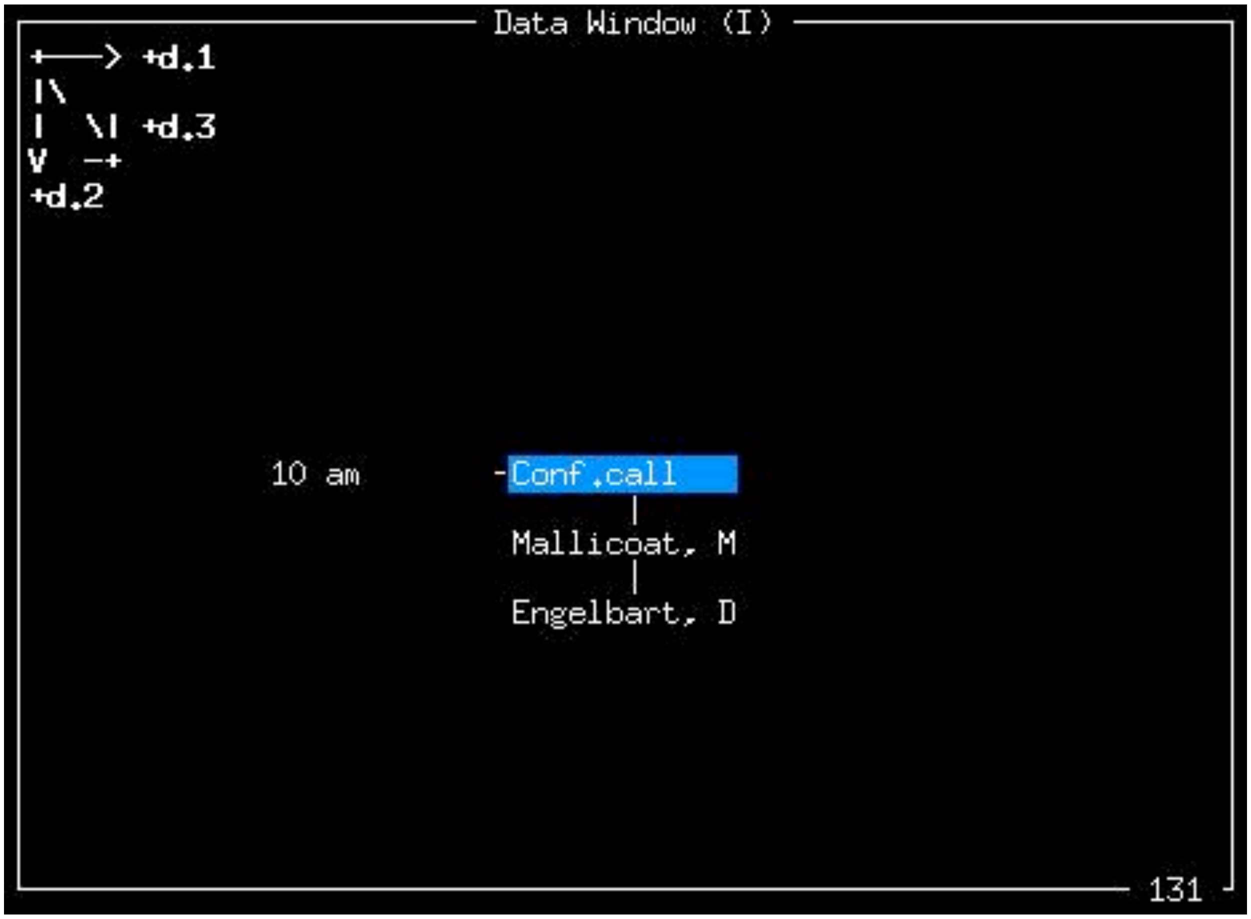
But this is no ordinary table, because it can open up as we move the cursor. When we move the cursor to the right, Tuesday's schedule appears.



```
──────────── Data Window (I) ────────────
+──> +d.1
|\
|  \| +d.3
V  -+
+d.2



         Apr 14  -Tuesday
                    |
                  10 am   -Conf.cal
                    |
                  12 noon -Lunch
                    |
                  6 pm    -Dinner

                                     117
```

This sudden change of scene can be a little startling, but that's one of the interesting aspects of the system. Each cell's connections are independent, and can be arranged into weavings that aren't visible till you get there. (There's much more to say about this, but no room to do it here.)

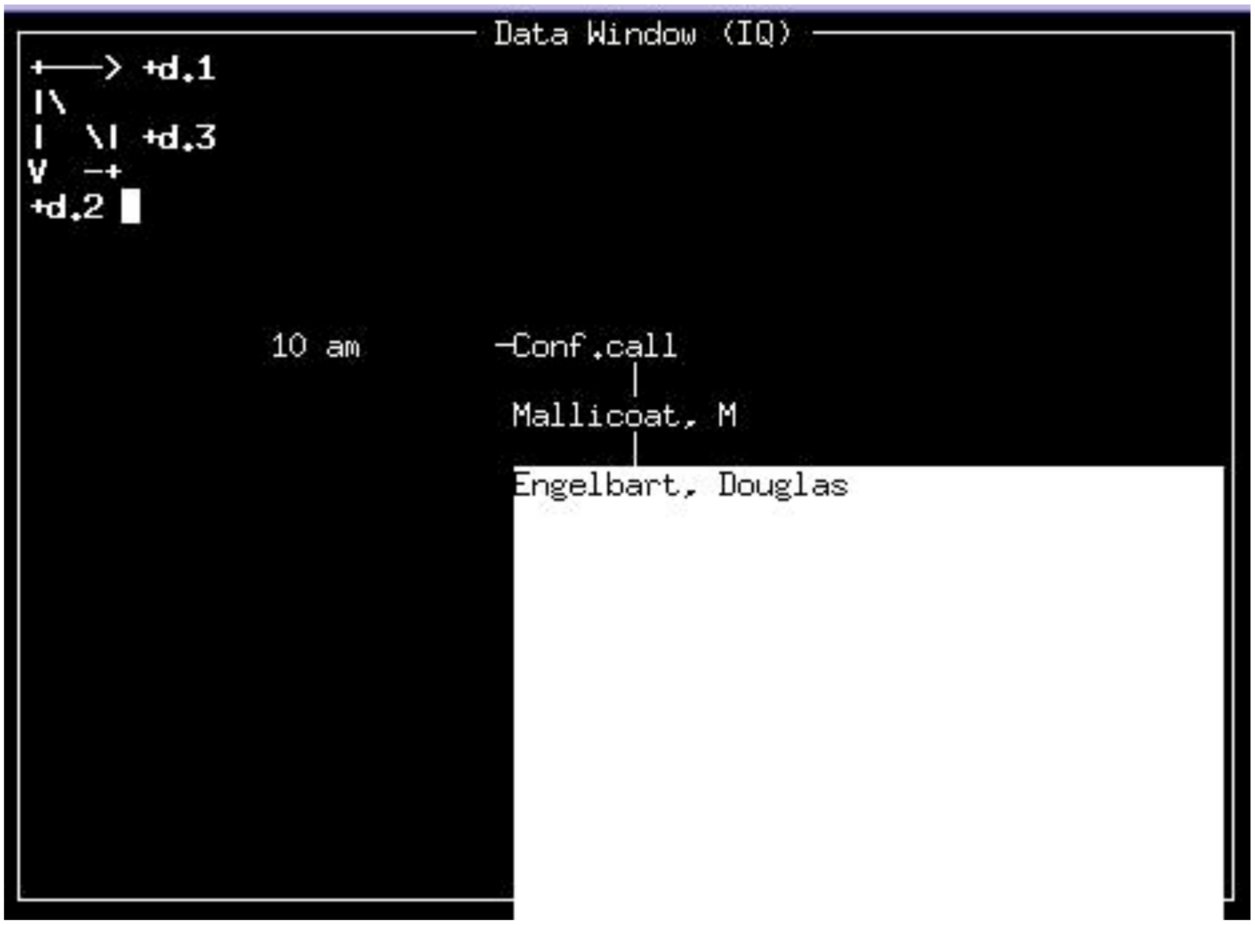
Let's step the cursor to see the details of the conference call mentioned on the screen. We step the cursor down and to the right, onto "Conf.call." (And let's widen the window a little.)



```
──────────────────────────── Data Window (I) ────────────────────────────
+──> +d.1
|\
|  \| +d.3
V  -+
+d.2




                10 am       -Conf.call
                                  |
                             Mallicoat, M
                                  |
                             Engelbart, D




────────────────────────────────────────────────────────────────── 131 ─
```

More cells open up below, showing the names of the people I'm planning to talk to in this conference call.

Now we move the cursor down to "Engelbart, D". And we hit a magical key to see the cell's contents. (The letter "q", for "quad window".)



```
Data Window (IQ)
+——> +d.1
|\
|  \| +d.3
V  -+
+d.2 █

            10 am       -Conf.call
                             |
                         Mallicoat, M
                             |
                         Engelbart, Douglas
```

This is just to show that cells can have lots more in them. The views you've been seeing are just a condensed way to see the structure. (Remember, this is just a prototype system, done as simply as possible).

A DIFFERENT APPROACH TO INFORMATION

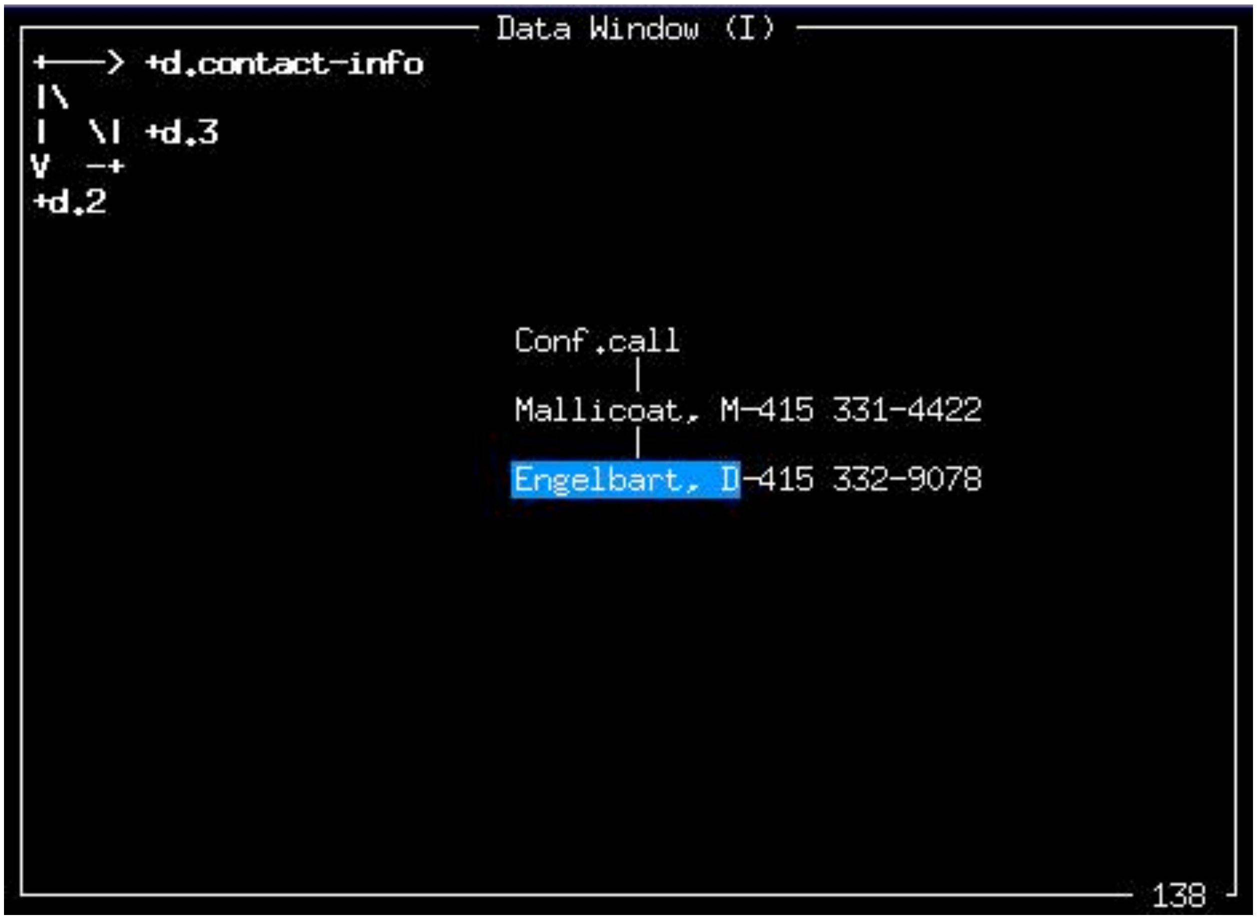
So far we've just seen hidden cells appear. But now let's look at what we can do with additional dimensions.

In the usual computer world, information is stored in so-called "databases", which are special-purpose resources separate from other parts of programs. (In some ways this makes sense, in other ways it doesn't.) But I want to show how in the ZigZag world we can integrate everything without special-purpose programs and facilities.

So now consider the conference call that's on the schedule-- where shall we put the telephone numbers that we need?

In most conventional systems, we'd have a separate database module that other programs would refer to. But here, instead, we can connect each item directly to the relevant information item. This is where the idea of "multiple dimensions" comes in.

We simply rotate the x axis into the ***contact-info*** dimension (as shown in the dimension map below):



```
Data Window (I)
+——> +d.contact-info
|\
|  \| +d.3
V  -+
+d.2

                    Conf.call
                         |
                    Mallicoat, M-415 331-4422
                         |
                    Engelbart, D-415 332-9078

                                                    138
```

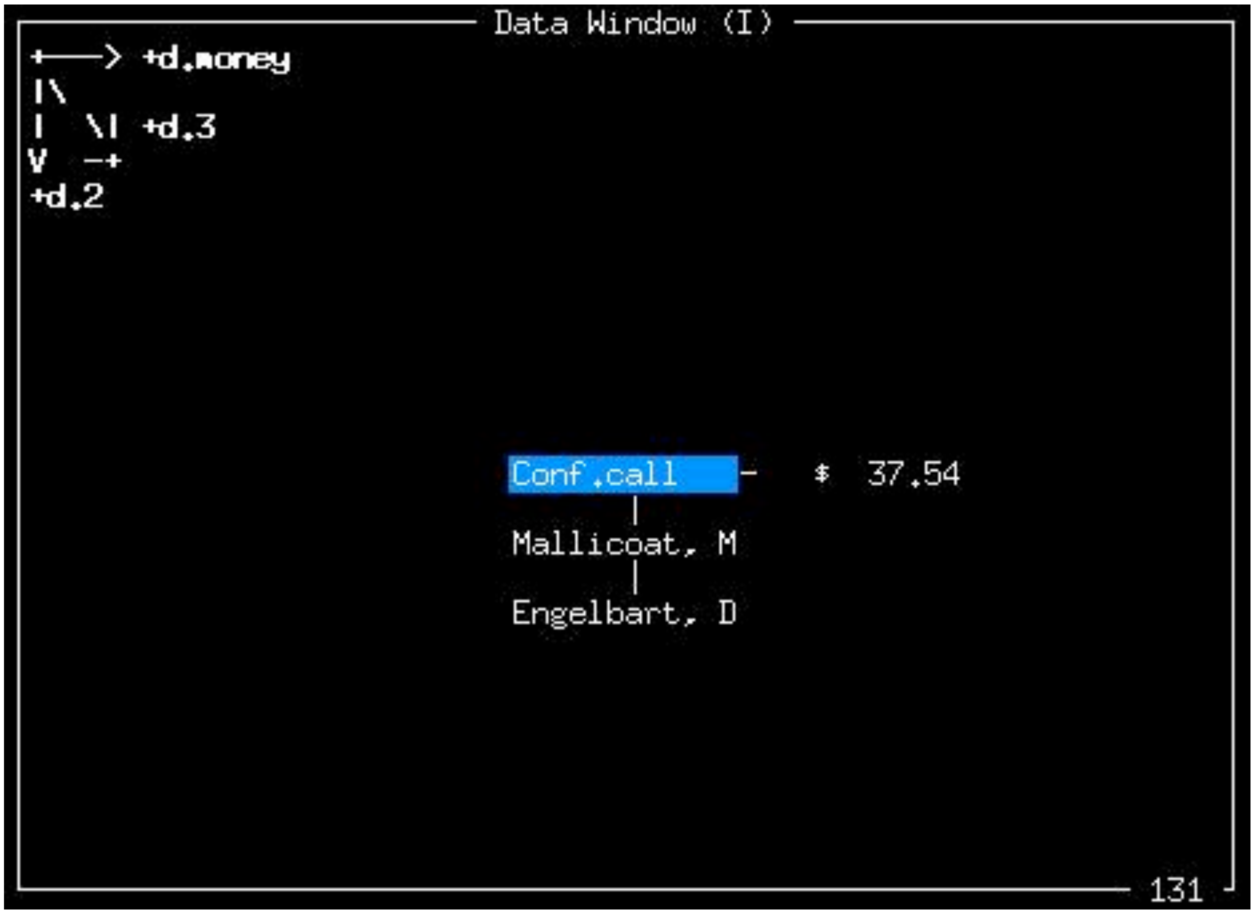
-- and there are the phone numbers, connected as they should be to the people whose numbers they are.

THE MONEY DIMENSION

Now here's another example.

How shall we keep track of expenses? For instance, where shall we take note of the cost of the conference call?

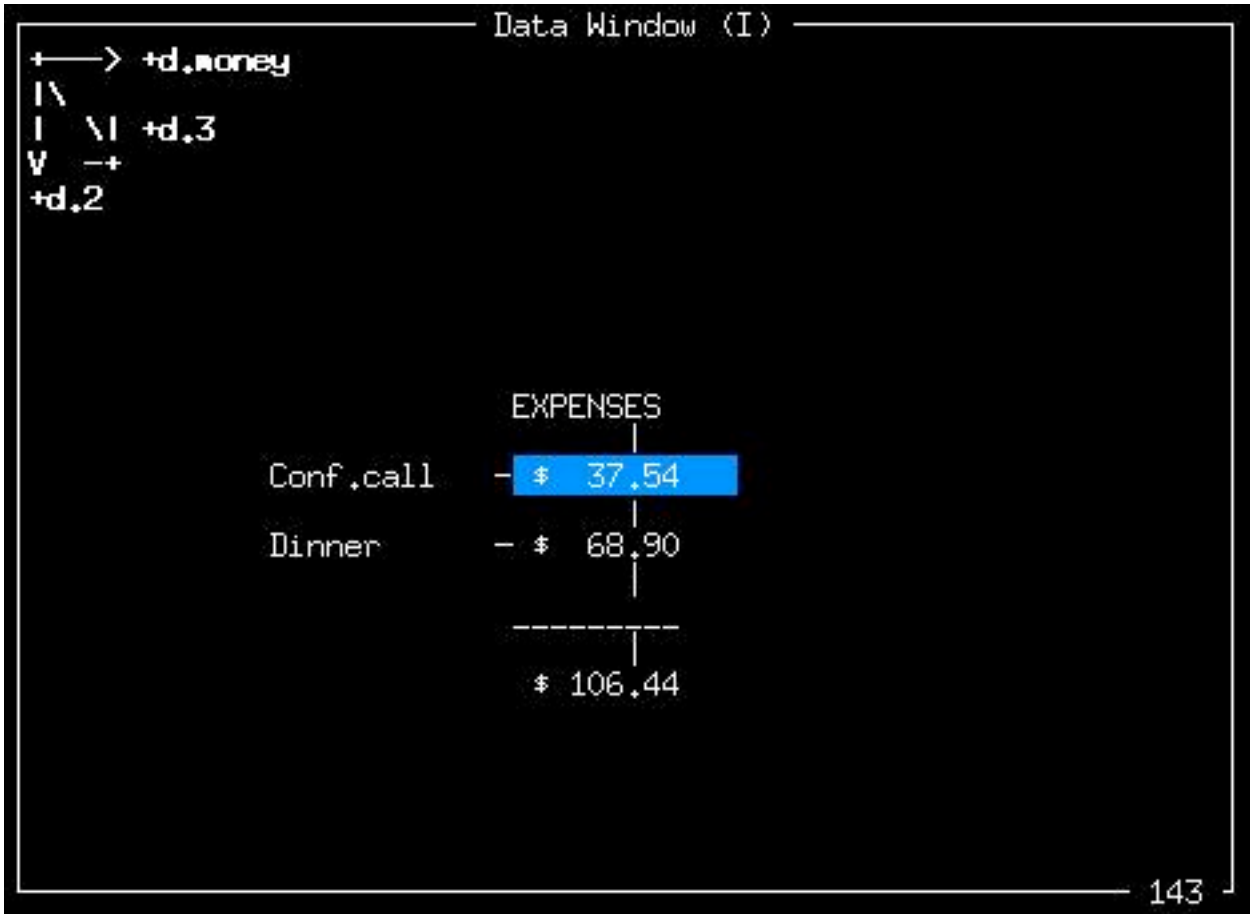
Let's go to the cell for the conference call and rotate the x-axis into the ***money*** dimension.



```
Data Window (I)
+---> +d.money
|\
|  \| +d.3
V  -+
+d.2

                    Conf.call   -    $  37.54
                         |
                    Mallicoat, M
                         |
                    Engelbart, D
                                                   131
```

That's where we have chosen to put the record of the cost of the conference call.

Where should we keep our other expenses for the day? Why, they too can be of course connected on the money dimension to the events whose cost they record. Just step to the right.



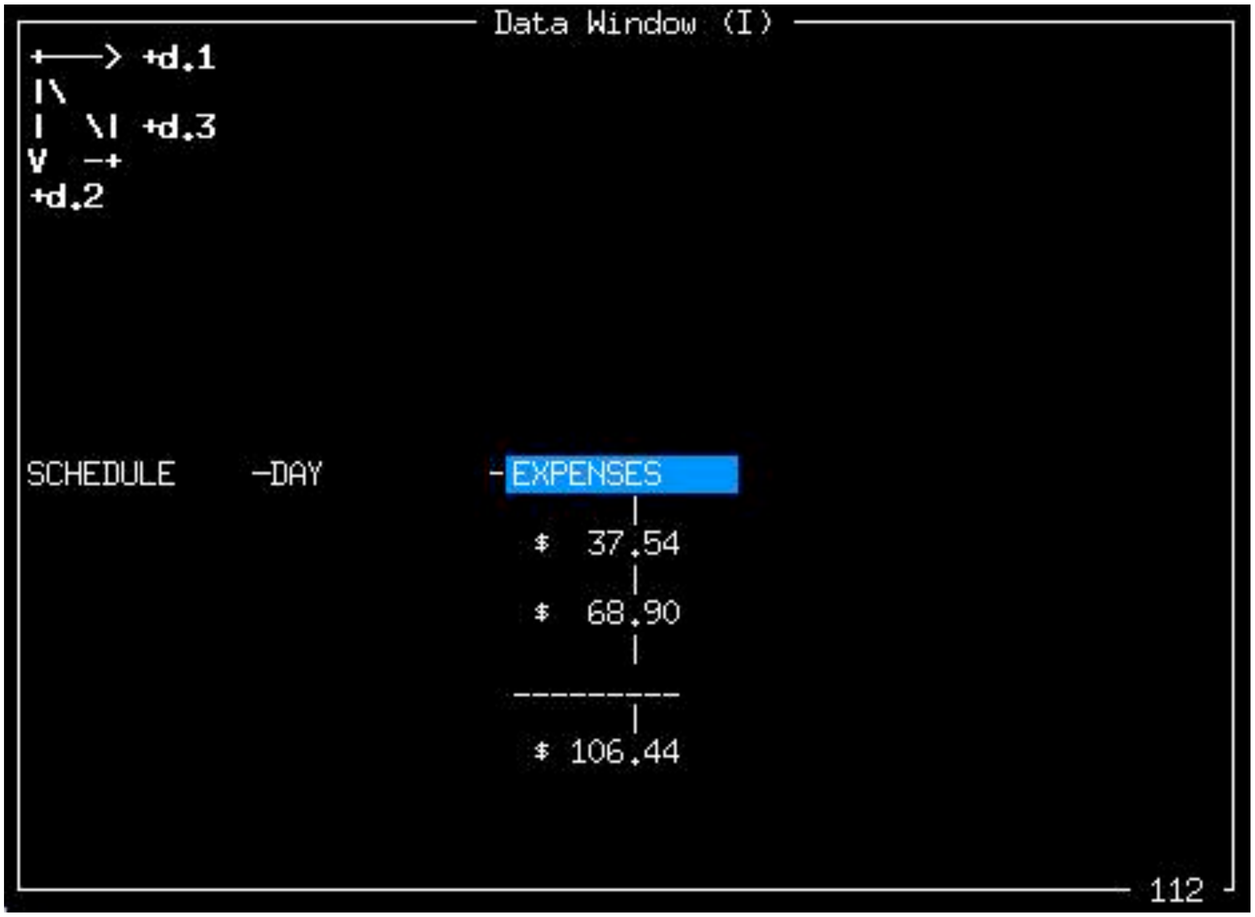
```
──────────────── Data Window (I) ────────────────
+──> +d.money
|\
|  \| +d.3
V  -+
+d.2


                    EXPENSES
                        |
     Conf.call     - $  37.54
                        |
     Dinner        - $  68.90
                        |
                   ---------
                        |
                     $ 106.44

                                              143
```

We see the other expenses right there connected to the cost of the telephone call-- along dimension 2 (d.2).

Now notice that the summation of the expenses (we've kept it simple-- just two items, for clarity--) is also along dimension 2.

Remember that all these dimensions are independent, so that any two dimensions can be seen at right angles to each other.

So dimension 2 won't change as we rotate the x-axis back from d.money to d.1.



```
Data Window (I)
+——> +d.1
|\
|  \| +d.3
V  —+
+d.2

SCHEDULE    -DAY       -EXPENSES
                             |
                       $  37.54
                             |
                       $  68.90
                             |
                       ---------
                             |
                       $ 106.44
                                               112
```

Well, look at that! As we rotated the x-axis back to dimension 1, dimension 2 (as shown on the y-axis) stayed the same, with the column of figures remaining visible and unchanged. The view of the expenses on d.2 did not change because we were only changing the x-axis.

Not only that, but we've returned to where we started. And when the x-axis finally showed d.1, the EXPENSES heading is-- guess where? Right in the same row on d.1 as the SCHEDULE heading, back in the top row, where this adventure began.

We've stepped the cursor up to the top row, where we started. If we were to move the cursor to the left, to the cell saying SCHEDULE, we could start back down and around again.

SUMMARY OF THE ZIGZAG SYSTEM

ZigZag is software built entirely out of connections in an interesting space, which I call Quantum Hyperspace (though it may have some other mathematical name I don't know about, since I stumbled upon it independently). It's best compared to beads on a string: every cell is like a bead with many holes, and you can put one string of any color (dimension) through each bead. (Note that it also generalizes my 1965 design of zipper lists.)

I don't know whether this space is of any interest to mathematicians, but I think it's interesting for software designers, for several reasons.

1. It offers a principled and consistent basis for interconnection-- not just random connections made up at the individual programer's whim, but chosen rationally by function. We can create rich conventions of interoperability between programs by creating dimensions with the specific functions we need.

2. ZigZag takes place in a conceptual space which can be made visible as rows and columns, and mapped to human activities and interests in a rich and insightful way. Thus it offers some very interesting interface possibilities. I see it as a possible new basis for a new world of software, outside of today's fads-- with no icons, no "metaphors" (dead-end scraps of comparison like the "desktop" and "clipboard"), and best of all, no "applications"-- separated, imprisoning, proprietary activity traps in which the commercial software vendors try to mire us.

3. Here's the punch line for wearable computers. ZigZag's row-and-column menus and multidimensional exploration may be especially suitable for wearable computers, since they allow the user to move easily in any two dimensions with four arrow keys-- indeed, to control two cursors at once with standard kids' game controls. And being able to spin in multiple dimensions can give us an unlimited universe to explore-- and create.

Space: the final frontier. Multidimensional space: the final frontier of the mind.

ACKNOWLEDGMENTS

I especially want to thank Keio University for its support and understanding; Marlene Mallicoat for her brilliant and unflagging encouragement, counsel and logistical support; Andrew Pam for his splendid implementation of the ZigZag prototype, and Bek Oberin for her recent contributions to its growth; Kazuhiro Yamada for helping restore the old "Porta-Xan" picture; and the Xanadu team over the years, a hundred-odd people who have supported these ideals and aspirations with their hearts, minds and very hard work.

BIBLIOGRAPHY

1. Nelson, Theodor Holm, "A File Structure for the Complex, the Changing and the Indeterminate." Proceedings of the ACM Twentieth National Conference, Pittsburg, Pennsylvania, 1965.

2. Nelson, Theodor Holm, ***Literary Machines***. Mindful Press, 1981 and later editions (current edition: 93.1. Distributed by Eastgate Systems, Cambridge MA (www.eastgate.com).

3. Nelson, Theodor Holm, ***Just Around the Corner: the ZigZag Book***. In preparation.

AVAILABILITY OF ZIGZAG(TM)

Shareware prototypes of the ZigZag program may be purchased at http://www.xanadu.net. We have two versions as of 98.06.24--

Difficult. The version exposed on the main page requires Linux to be pre-installed.
Easy. There is also (currently hidden) a downloadable boot floppy from which a standard Wintel computer may be rebooted into ZigZag. The package will be downloaded automatically if you go to http://www.xanadu.net/zigzag/zzdemo.zip. This package must then be unzipped (you will need an Unzip module, such as WinZip, available "for evaluation" at http://www.winzip.com) and moved to a floppy using Rawrite (included in the package). Rebooting will then bring up ZigZag.