Visualizing E-mail with a Semantically Zoomable Interface

Ellen Diep*
Tufts University

Robert J. K. Jacob†
Tufts University

Abstract

We introduce a semantically zoomable interface that displays e-mails as interactive objects rather than files containing lines of text, as in traditional e-mail interfaces. In this system, e-mails are displayed as node objects called *e-mail nodes* within a 2.5-dimensional world. The e-mail nodes are semantically zoomable and each may be rearranged to different locations within the plane to organize threads, topics, or projects. The prototype for this system was built using the Piccolo toolkit, the successor of Pad++ and Jazz [2, 3].

Keywords: Clutter, e-mail, information visualization, Piccolo, semantic zoom

1 Motivation

Figure 1: Example of a traditional e-mail inbox.

In a typical e-mail interface, we may find a set of e-mails listed in a sorted, linear format (Figure 1). Although it is a format that we have seen since our first encounter with an e-mail application, many of us still find it difficult to efficiently organize e-mails or quickly find what we need amidst the clutter. With the large amount of personal and business e-mails – sent, received, forwarded, carbon-copied, or spammed – each day, this has increasingly become a problem for many e-mail users.

Interface developers have tried to work around the problems of e-mail clutter by adding feature enhancements to e-mail interfaces. Feature enhancements include sort functions, search functions, folders, etc, which all have their benefits but not without their trade-offs. For example, studies show that most users actually end up keeping all of their e-mail in one large folder [9]. Some of these feature enhancements and others have become distractions to users rather than serving to efficiently manage e-mails. Many users still choose to manually search long lists of e-mails to find the target messages rather than use overloaded features in their e-mail applications. Thus, linear representations of e-mail often cause clutter and confusion for the user, and feature enhancements have turned out to be only add-ons, not solutions to the clutter problem.

To reduce task management issues and clutter within e-mail systems, we need to look for other ways of designing e-mail interfaces. Rather than focusing on feature enhancements, we need to focus more on natural designs that support user interaction. Research and applications investigating zoomable user interfaces (ZUIs) have been produced for news articles [7], small-display PDA calendars [1], GIS maps and other graphics. Our system presents e-mails as semantically zoomable objects within a ZUI. If e-mails are represented as if they were tangible objects, rather than static or linear objects, we hope user interaction with them will be more natural and intuitive.

2 The Zoomable E-mail System

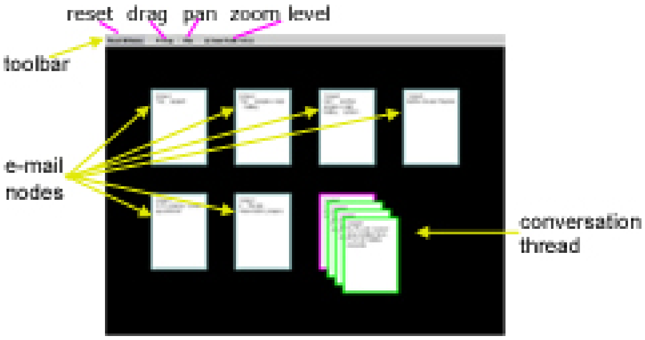


Figure 2: E-mail interface prototype.

Our prototype system (Figure 2) displays e-mails as semantically zoomable, rectangular nodes. The toolbar allows the user to select “drag” or “pan” radio buttons to drag the nodes or pan the entire viewing area, respectively. It also has a reset button to reset the nodes to their previously saved positions and a zoom scale indicator that shows the current zoom level.

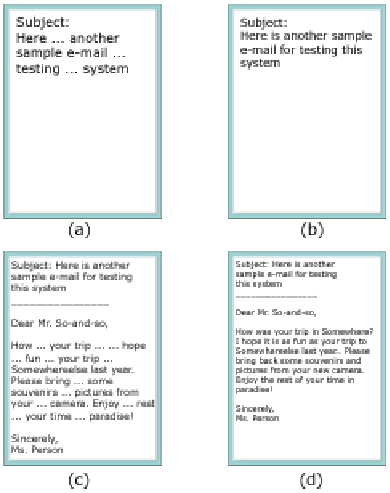
2.1 Design Rationale

This interface aids in reducing clutter by allowing the user to develop his or her individual method(s) of organizing e-mail and moving messages around as if they were physical objects on a surface. In this system, e-mails are displayed as node objects called *e-mail nodes* within a 2.5-dimensional world (2D plane + spatial zooming) (see Figure 2). The user may rearrange each e-mail node to different locations within the plane by clicking and dragging the nodes with the mouse. By dragging the e-mail nodes to different locations in the vast 2D plane, the user may stack and group the messages in whichever way that is intuitive and convenient, thereby reducing clutter and confusion. This allows the user to have total control over the physical placement of e-mail objects. The user may create his or her individual mnemonic or mental model of where the e-mails belong or use a standardized method of organizing their e-mails.

*e-mail: ellen.diep@tufts.edu
†e-mail: jacob@cs.tufts.edu

2.2 Semantic Zoom with E-mail

The e-mail nodes are also semantically zoomable. Contents displayed within the nodes varies depending on the zoom scale. The user may zoom out for an abstract view of the e-mail contents, zoom in for more detail of the e-mail contents, or zoom all the way in for the full contents of the message.



Subject:
Here .... another
sample e-mail ...
testing ... system

Subject:
Here is another sample
e-mail for testing this
system

Subject: Here is another
sample e-mail for testing
this system

Dear Mr. So-and-so,

How ... your trip .... hope
... fun .... your trip ...
Somewhereelse last year.
Please bring .... some
souvenirs .... pictures from
your .... camera. Enjoy .... rest
... your time .... paradise!

Sincerely,
Ms. Person

Subject: Here is another
sample e-mail for testing
this system

Dear Mr. So-and-so,

How was your trip in Somewhere?
I hope it is as fun as your trip to
Somewhereelse last year.. Please
bring back some souvenirs and
pictures from your new camera.
Enjoy the rest of your time in
paradise!

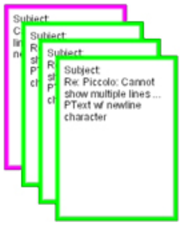
Sincerely,
Ms. Person

Figure 3: An e-mail node viewed at different zoom scales: (a) zoom level 1; (b) zoom level 2; (c) zoom level 3; (d) zoom level 5.

As the user zooms out, "unimportant" words, such as "the," "a," adverbs, and adjectives are ellided depending on the zoom level. See Figure 3. At zoom level 1.0, only a summarized subject line is displayed. At zoom level 2.0, the entire subject line is displayed. At level 3.0, the subject and a summarized version (sans-adjectives/adverbs) of the body are displayed. At zoom level 4.0, the subject and a less summarized version (sans-adverbs) of the body are displayed. At levels 5.0 and up, the entire header (sender, recipient, date, and subject information) and the entire body of the e-mail are displayed. This ad hoc method of summarization and could be improved by using summarization heuristics like those proposed by Lam et al. [6].

When zooming out, the same semantics are applied to the associated zoom scale. The prototype also automatically adjusts the font size based on the zoom scale using the formula: size = c1 / (scale + c2), where size is the resulting font size, c1 and c2 are positive, nonzero constants, and scale is the variable zoom scale [4]. As the zoom level decreases below 1.0, however, the nodes shrink and the text become illegible, giving a "bird's eye-view" of the nodes on the 2D plane.

2.3 Conversation Threads



Subject:
Re: Piccolo: Cannot
show multiple lines ...
PText w/ newline
character

Figure 4: A color-coded conversation thread. From top-left to bottom-right, the first e-mail node is the original message (magenta border), followed by three subsequent replies (green border) to the original message.

This system incorporates a color scheme for conversation threads. In Figure 4, the e-mail node with the magenta border is the original message, and the e-mail nodes with green borders are the replies to the original message. Conversation threads are stacked from oldest to newest by send date, with the most recent e-mail node at the top. In future implementation of this system, this can be done automatically as e-mails arrive. Rohall and Gruen also propose using visual cues with graphics and color schemes for displaying e-mails and threads of e-mails [8].

2.4 Implementation Using Piccolo

The Piccolo toolkit, which is the successor of Pad++ and Jazz, was used to build the prototype of this system [2, 3]. Piccolo was chosen for its reputation as a well-researched and documented ZUI framework, for its ease of use, and for its smooth animation and transitions between zooms. A Java swing toolbar was added for resetting nodes to their original position and scale, choosing pan or drag interaction methods, and displaying the zoom scale. Scrollbars were also added for scrolling to nodes that are not within the bounds of the current screen display.

The system employs the default zoom event handlers of Piccolo. Studies have shown that users experience less spatial disorientation and more confidence in their actions when zoom movement is constrained [5]. We therefore set the minimum zoom scale threshold to 0.1 (10% of the original size) and the maximum zoom to 7.0 (700% of the original size).

3 Future Directions

Our prototype system supports our basic design concepts of the "semantically zoomable e-mail system." The next step is to conduct a user study to evaluate this approach. Extensions include using proven summarization heuristic algorithms, saving of workspace information, and automatic organization into conversation threads. This last must be done carefully, to avoid "overloading" the system with excessive features that could detract from the simplicity of the basic design approach.

References

[1] B. Bederson, A. Clamage, M. Czerwinski, and G. Robertson. A fisheye calendar interface for PDAs: providing overviews for small displays. *ACM CHI Extended Abstracts,* pp. 618-619, 2003.

[2] B. Bederson and J. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. *ACM UIST Symposium on User interface software and technology*, pp. 17-26, 1994.

[3] B. Bederson, J. Meyer, and L. Good. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *ACM UIST Symposium on User interface software and technology*, pp. 171-180 2000.

[4] L. Good, M. Stefik, P. Baudisch, and B. Bederson. Automatic text reduction for changing size constraints. *ACM CHI Extended Abstracts*, pp. 798-799, 2002.

[5] S. Jul. "This Is a Lot Easier!": Constrained Movement Speeds Navigation. *ACM CHI Extended Abstracts*, pp. 776-777, 2003.

[6] D. Lam, S. Rohall, C. Schmandt, and M. Stern. Exploiting E-mail Structure to Improve Summarization. *ACM CSCW Conference on Computer Supported Cooperative Work*, 2002.

[7] E. Rennison. Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes. *ACM UIST Symposium on User interface software and technology*, pp. 3-12, 1994.

[8] S. Rohall and D. Gruen. ReMail: A Reinvented Email Prototype. *ACM CSCW Conference on Computer Supported Cooperative Work*, 2002.

[9] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. *ACM CHI Conference on Human factors in computing systems,* pp. 276-283, 1996.