

Appendix A: Scenarios

Snap-Together Visualization has been used with a variety of data and visualizations that demonstrate its breadth and usefulness. Example applications include: WestGroup case law, Census Bureau statistics, GIS maps, Maryland State Highway Administration incident data, personal photo libraries, stock market portfolios, web-site logs, mailing address databases, technical-report databases, and hierarchical file structures. These scenarios use a variety of data types, including textual, numeric, geographic, hierarchical, and image. They also employ a variety of visualizations including commercial and non-commercial, and Windows-based as well as web-based. Each scenario includes the specification for the coordinated-visualization interface using the notation of the Snap model from Chapter 3.

A.1 Web-Site Logs

In related work on visualizing web-site logs, Hochheiser [HS99] created scripts to parse web-site log files into an Access database. These files contain data about hits to the HCIL web site. Using Snap, a coordinated-visualization interface (Figure A.1) was easily constructed for examining what other web pages refer many readers to pages on the HCIL web site. The three visualizations at the top (outliner, Treemap, Internet Explorer) form a site browser for the HCIL web site. The outliner and Treemap display the hierarchical structure of the site. Selecting a page in either displays that page in IE.

The Treemap shows that the HCIL home page, Pad++, and the Visible Human Explorer are the most frequently visited pages.

The two visualizations at the bottom (scatter plot, and IE) display other pages that refer readers to the selected page in the site browser. The plot shows referring pages along the X-axis and the number of hits referred (during October 1998) on the Y-axis. Selecting the most frequent referrer (110 hits) to the HCIL home page reveals Human Factors International in IE. Exploring reveals other common referrers, including Ben Shneiderman's page, the Department page, and Yahoo's HCI institutes page. Selecting the Visible Human Explorer page in the outliner shows nearly 1000 hits from the National Library of Medicine page. Selecting to open this page indeed reveals a prominent link to the HCIL page. Naturally, HCIL lab members explored to discover referrer patterns to their personal pages.

The Snap specification for this interface is:

$$\text{Visualizations} = \{ (\text{outliner}, \text{pages}), (\text{Treemap}, \text{pages}), (\text{IE}_{\text{top}}, \text{pages}), \\ (\text{plot}, \text{pageReferrers}), (\text{IE}_{\text{bottom}}, \text{pageReferrers}) \}$$
$$\text{Coordinations} = \{ ((\text{outliner}, \text{select}), (\text{Treemap}, \text{select})), \\ ((\text{outliner}, \text{select}), (\text{IE}_{\text{top}}, \text{load-PK})), \\ ((\text{outliner}, \text{select}), (\text{plot}, \text{load-FK}_{\text{page}})), \\ ((\text{plot}, \text{select}), (\text{IE}_{\text{bottom}}, \text{load-PK})) \}$$

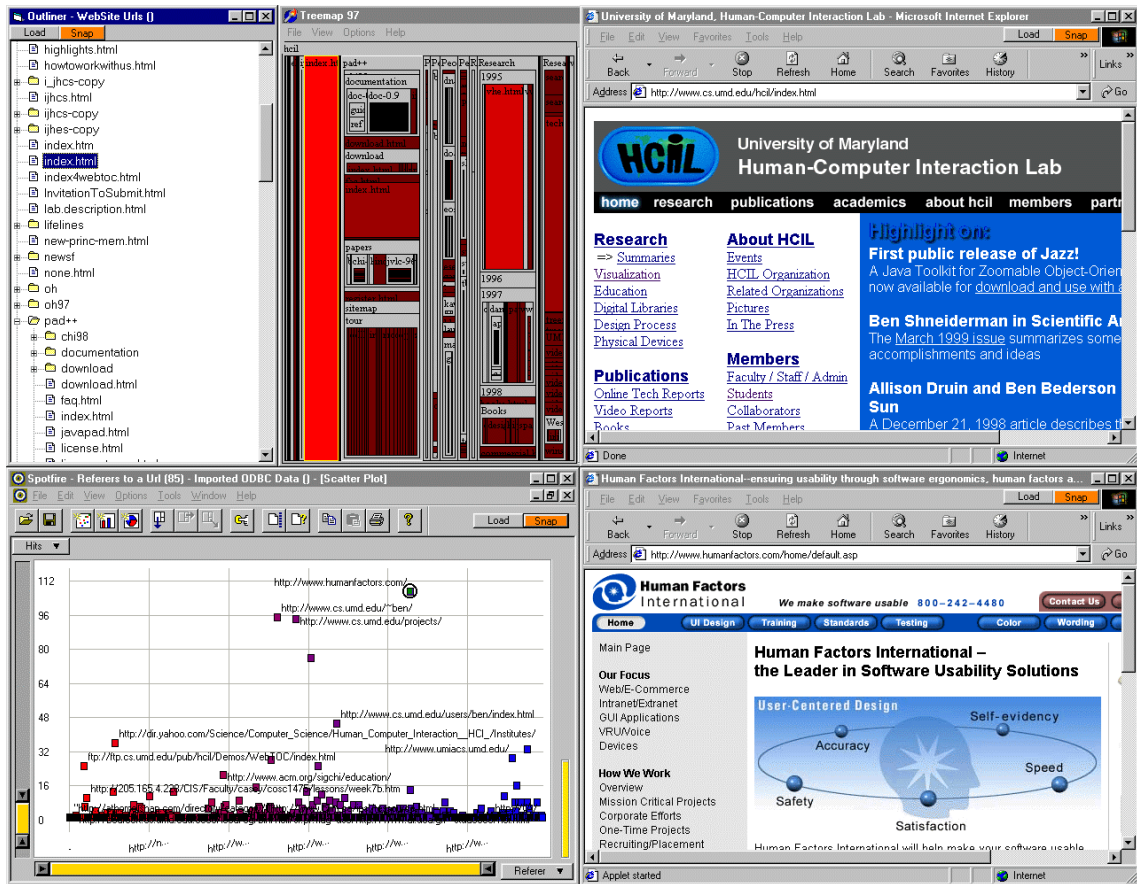


Figure A.1: Web-site logs scenario

A.2 Census Data

Figure A.2 is an interface for exploring Census population data of U.S. states (left) and counties (right). Users can explore from nominal, geographic and numeric perspectives. Selecting Maryland reveals that it ranks very high in terms of income per capita and percent college graduates. Maryland has two counties that have much higher percentage of college graduates than the others. One of these, Montgomery County, has the highest per capita income and is clearly located just north of DC.

This example demonstrates the use of ESRI MapObjects, a component of the popular ArcView GIS software.

Snap is in use at the Census Bureau to prototype user interfaces for CD-ROM products. Census analysts have also found the capability to relate data between maps and plots extremely helpful. Continued work on Snap at Census has already enabled multiple selection for brushing and linking, and connection to intranet-based Oracle database servers.

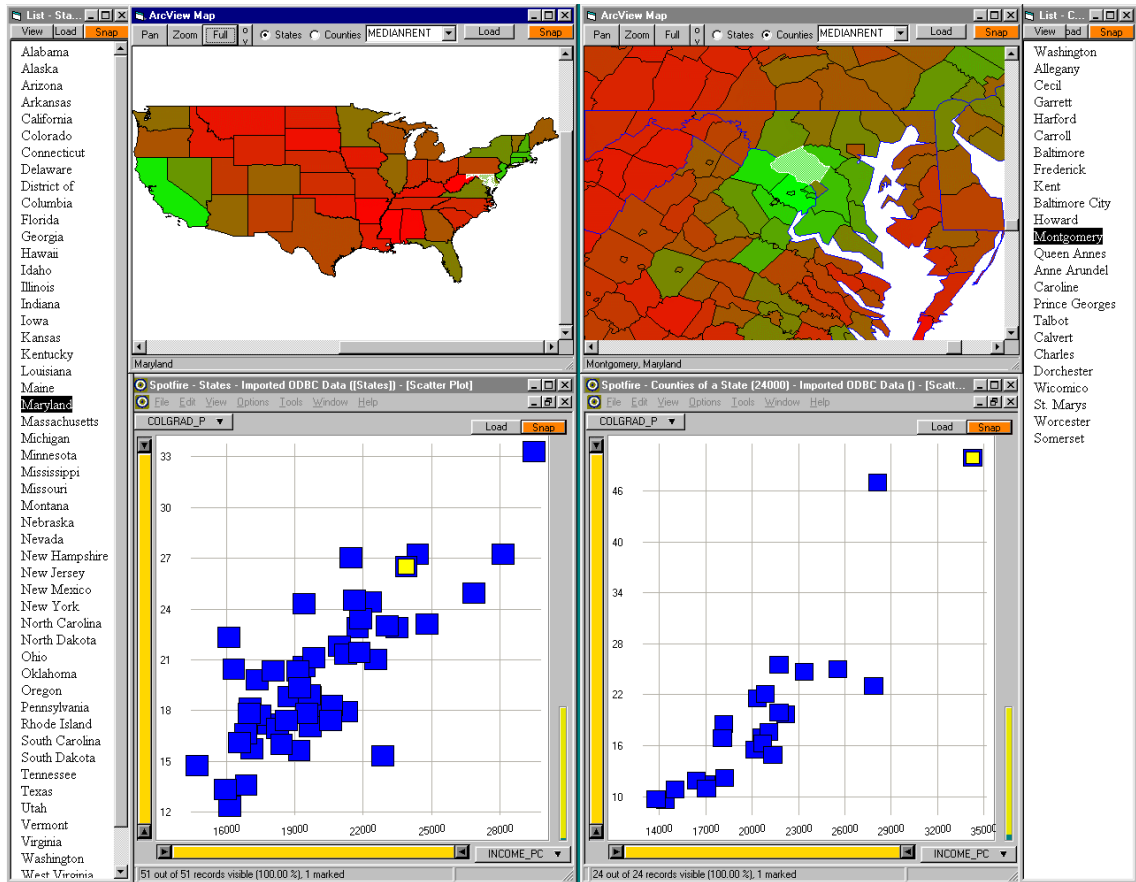


Figure A.2: Census data scenario

The Snap specification for this interface is:

$$\text{Visualizations} = \{ (\text{map}_{\text{states}}, \text{states}), (\text{plot}_{\text{states}}, \text{states}), (\text{list}_{\text{states}}, \text{states}), (\text{map}_{\text{counties}}, \text{counties}), (\text{plot}_{\text{counties}}, \text{counties}), (\text{list}_{\text{counties}}, \text{counties}) \}$$

```

Coordinations = { ((mapstates, select), (plotstates, select)),
                  ((mapstates, select), (liststates, select)),
                  ((mapcounties, select), (plotcounties, select)),
                  ((mapcounties, select), (listcounties, select)),
                  ((mapstates, select), (mapcounties, zoom)),
                  ((mapstates, select), (plotcounties, load-FKstate)),
                  ((plotcounties, load-FKstate), (listcounties, load-FKstate)) }

```

A.3 Photo Libraries

Snap was used in an HCIL research project on user interfaces for personal digital-photo libraries [KTS00] to explore many possible designs. The lab has accumulated a database of scanned photos of lab members and activities spanning 10 years. It includes annotations such as members' names, dates, locations, and other information.

In Figure A.3, a thumbnail visualization shows a collection of a few hundred photos. The scatter plot displays a time-line overview of the photos, with date on the X-axis and members' names on the Y. Users can see trends and patterns. For example, vertical stripes of dots represent group events, pictures of many members on the same date. The large stripe in the middle is many photos from the 1992 HCIL Open House. Selecting a photo from winter '89 displays the full-size photo from a ski trip, a list of names of members in the photo, and details of photo attributes.

Other interface variations include locating photos by members' names or locations, selecting a person in a photo to find other pictures of that person, etc.

The Snap specification for this interface is:

```

Visualizations = { (thumbnails, photos), (plot, photos), (IE, photos),
                  (listpeople, appearances), (listdetails, photos) }

```

```

Coordinations = { ((thumbnails, select), (plot, select)),
                  ((thumbnails, select), (IE, load-PK)),
                  ((thumbnails, select), (listpeople, load-FKphoto)),
                  ((thumbnails, select), (listdetails, load-PK)) }

```

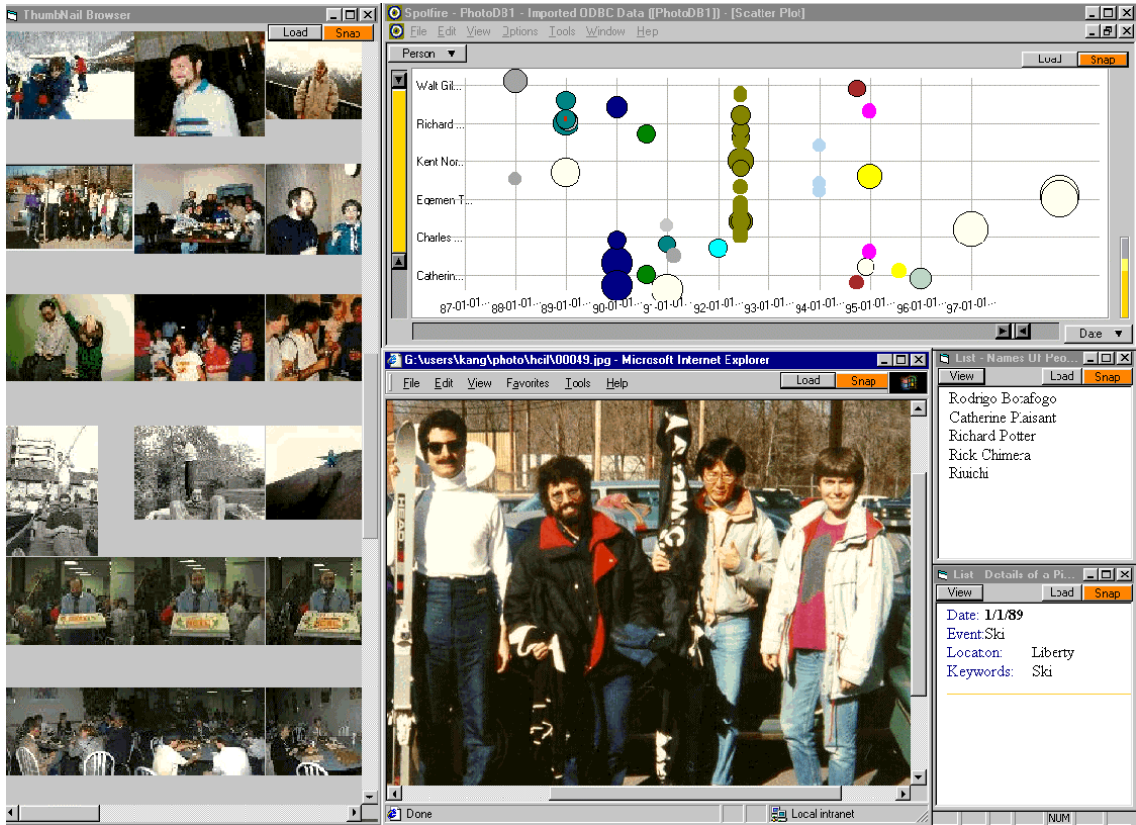


Figure A.3: Photo libraries scenario

A.4 WestLaw Case-Law Documents

Significant inspiration for the Snap concept resulted from an HCIL research project for WestLaw on visualization of case-law documents. Snap was used to prototype work-benches for legal analysts. A major task that the analysts perform is to search large case-law document databases using keywords, and then examine resulting cases for relevance to a current case.

This user interface in Figure A.4 is for browsing search results. The visualizations at the top of the screen display the hits resulting from a search. The Snap search

window is used to enter search terms (described in Chapter 4). The hits are displayed both textually and graphically by date and search relevance. Selecting a case displays it in the case viewer at the bottom of the screen.

The case viewer displays the text-intensive details of the case in a manner that supports rapid navigation. A case is composed of a judge's decision text, which is partitioned into sections. Each section has a WestLaw headnote, containing a categorization and annotation. WestLaw's existing user interface simply listed out all the information in a single web page with many intra-links between sections and headnotes. Since users often refer to headnotes while browsing the decision text, yet need to scan the decision as a contiguous text, a two-frame synchronized-scrolling approach is more appropriate. The main list visualization on the right displays the text of the case by sections. The center list displays WestLaw headnotes for each section, and synchronizes scrolling with the main text. Since many cases are long, containing 10 to 50 headnotes, users can quickly jump to a section by selecting section numbers from the overview list on the left.

This example demonstrates how Snap would be ideal for rapid web-based user interface construction. In fact, based on this prototype, WestLaw did implement this case-viewer design in their web site (www.westlaw.com).

The Snap specification for this interface is:

Visualizations = { (search, phrases), (list_{hits}, cases), (plot, cases),
 (list_{overview}, sections), (list_{headnotes}, sections),
 (list_{text}, sections) }

Coordinations = { ((search, search), (list_{hits}, load-FK_{search})),
 ((list_{hits}, load-FK_{search}), (plot, load-FK_{search})),
 ((list_{hits}, select), (plot, select)),
 ((list_{hits}, select), (list_{overview}, load-FK_{case})),

```

((listoverview, load-FKcase), (listheadnotes, load-FKcase)),
((listoverview, load-FKcase), (listtext, load-FKcase))
((listoverview, select), (listheadnotes, scroll)),
((listheadnotes, scroll), (listtext, scroll)), }

```

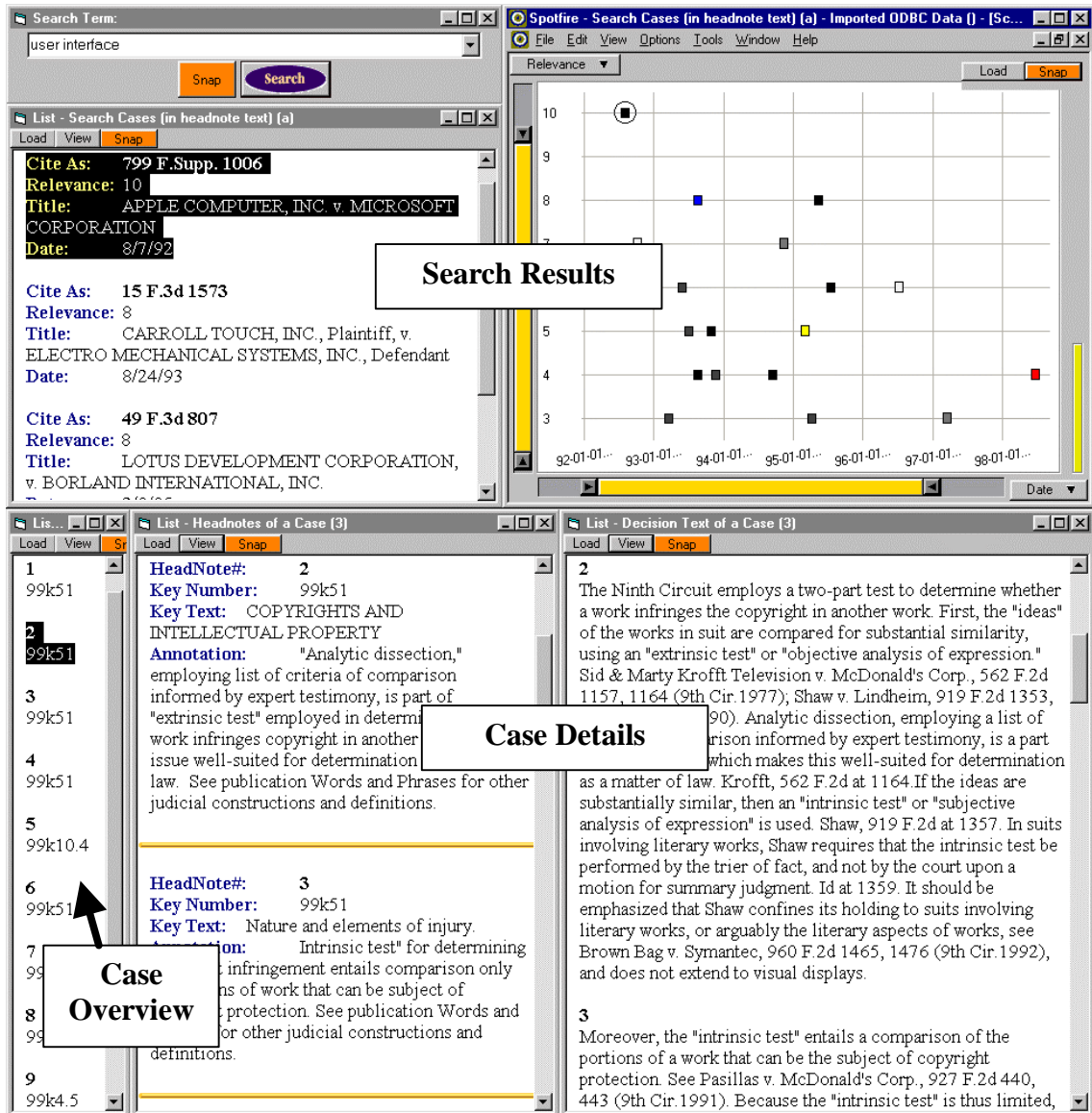


Figure A.4: Case-law scenario

A.5 Highway Incident Data

For an HCIL research project on visual aggregation strategies, Fredrikson [FNP99] used Snap with Maryland State Highway Administration incident data. She identified temporal, geographical and categorical attributes as ideal candidates for aggregation. The drill-down coordination was used to allow users to select aggregates in one visualization to display aggregate contents in another visualization. For example, Figure A.5 displays aggregations of highway accidents by day of the week in a bar chart. Selecting Monday, which had the most accidents, reveals the locations of individual accidents on the road map of the Baltimore, MD area.

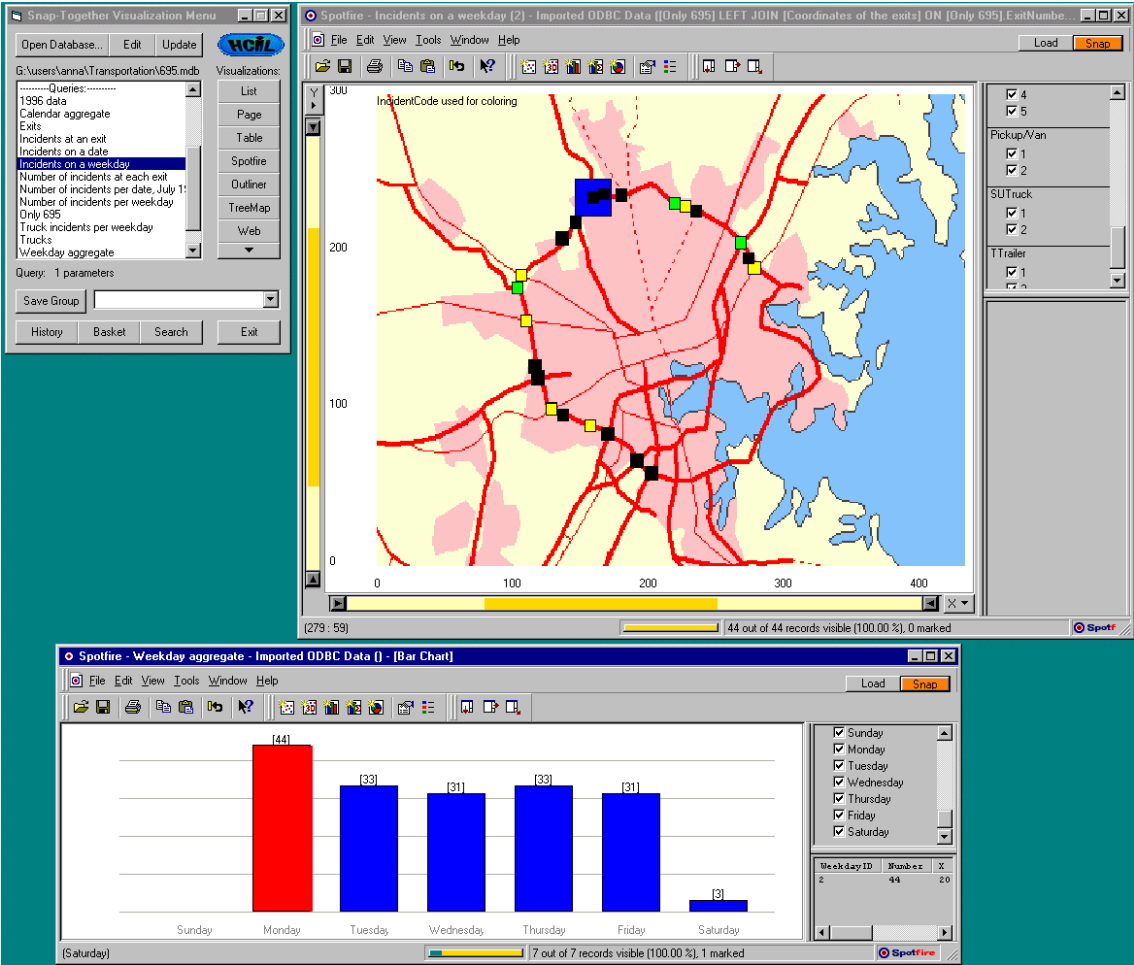


Figure A.5: Highway incident data scenario

While the data set used in this project was not large (~1000 tuples), this technique demonstrates how Snap can be used to explore very large-scale relations using drill down. For example, 1,000,000 traffic incidents could be aggregated into 1,000 aggregates, each with 1,000 incidents. This could be displayed with two coordinated visualizations, an overview of 1,000 points, and a detail view of 1,000. Furthermore, this approach can be repeated by chaining several visualizations, adding an additional visualization for each level to multiply by powers of 1,000.

The Snap specification for this interface is:

```
Visualizations = { (barchart, dayAggregates), (map, incidents)      }
Coordinations = { ((barchart, select), (map, load-FKday))      }
```

A.6 Mailing Address Database

In Figure A.6, Snap is used to explore addresses in the HCIL mailing-list database. The names are displayed in a simple table. The table is coordinated to IE, in which a query is loaded that formulates a mailing address as a URL query string to Yahoo Maps. Then, selecting a name in the mailing list displays a map of the location of that address. This example demonstrates how web services such as Yahoo Maps can be used as snap-able visualizations.

The Snap specification for this interface is:

```
Visualizations = { (table, addresses), (IE, addressQueryStrings)  }
Coordinations = { ((table, select), (IE, load-PK)) }
```

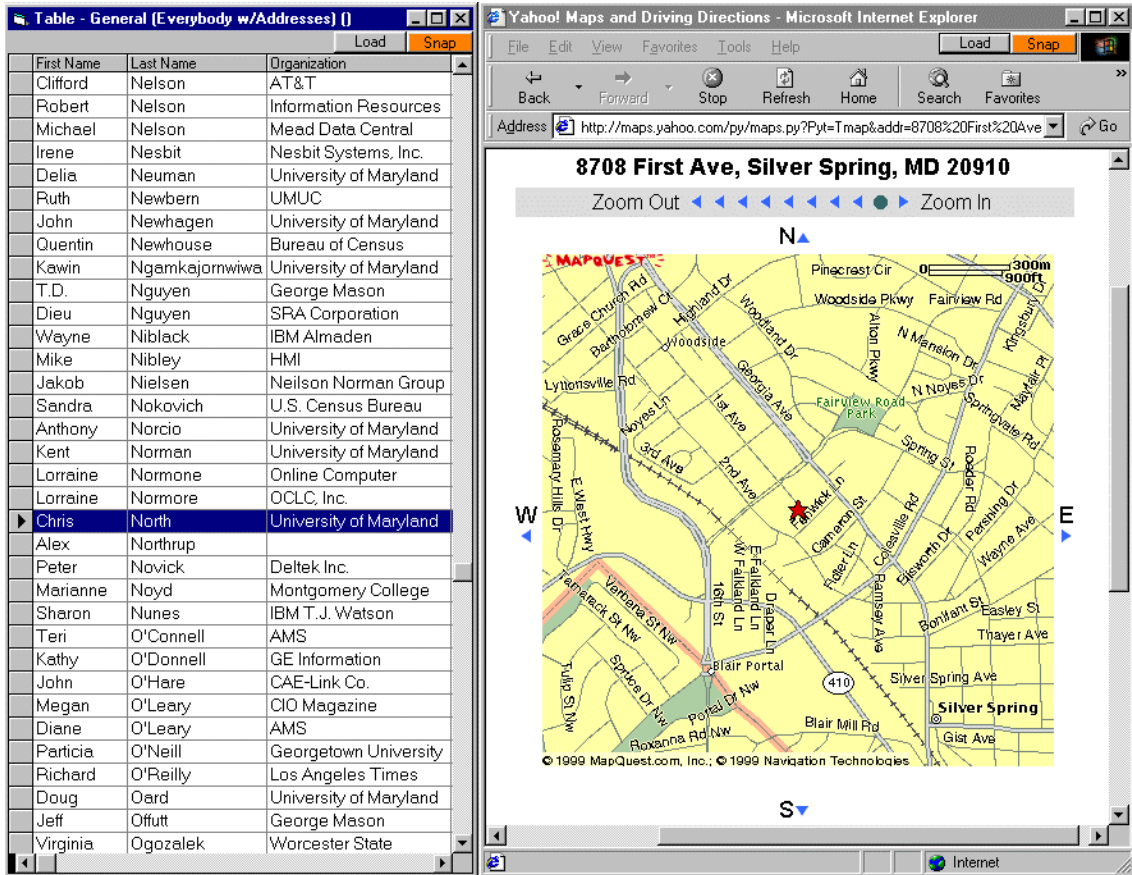


Figure A.6: Mailing address database scenario

A.7 Files and Folders

The file-folders scenario in Chapter 1 demonstrates how Java applets such as Hyperbolic Trees can be snap-enabled using IE (Figure A.7). IE is also useful as a general-purpose file viewer for images, HTML, PDF and Word documents, etc.

The Snap specification for this interface is:

Visualizations = { (plot, folders), (hyperbolic, folders), (table, files), (IE, files) }

Coordinations = { ((plot, select), (hyperbolic, select)), ((plot, select), (table, load-FK_{folder})), ((table, select), (IE, Load-PK)) }

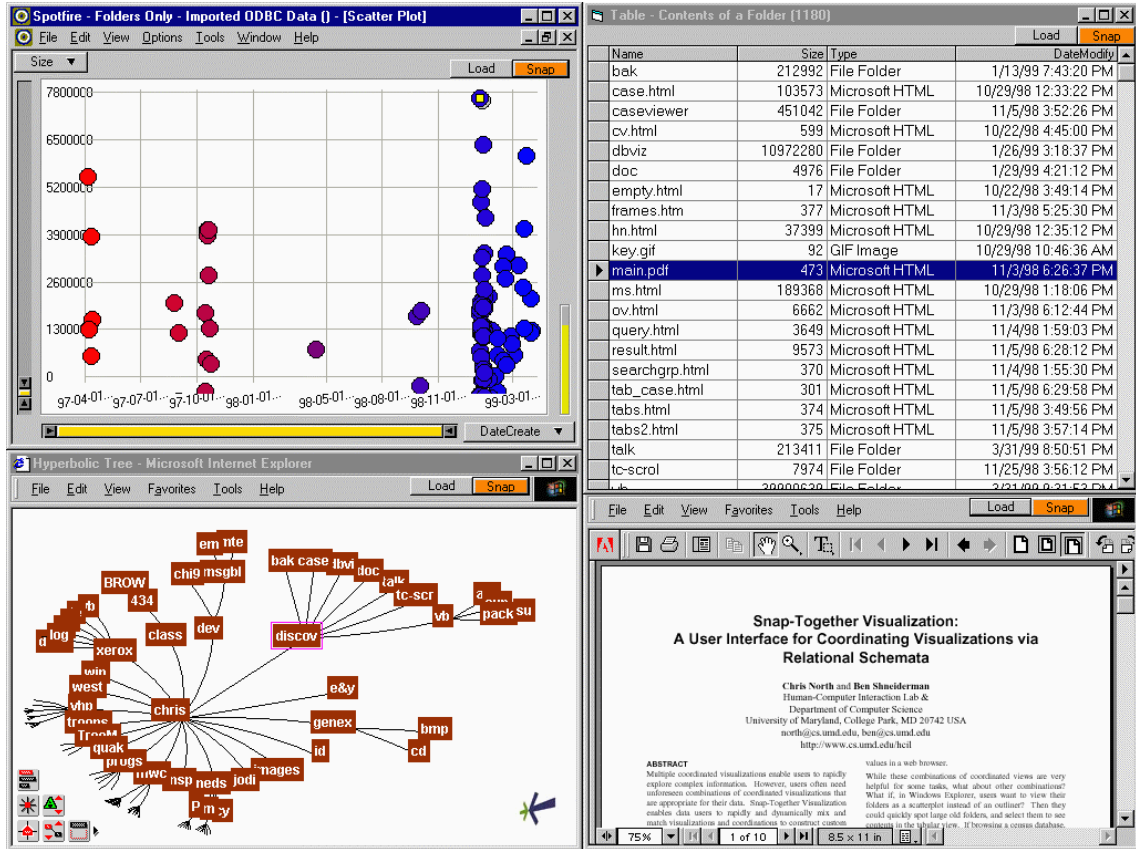


Figure A.7: Files and folders scenario

A.8 Stock Market Portfolios

In data analysis, it is often useful to view both the graphical visualization as well as the detailed numeric spreadsheet. In Figure A.8, Snap is used to display a financial stock portfolio. Brushing and linking relate the Treemap and spreadsheet.

The Snap specification for this interface is:

Visualizations = { (treemap, stocks), (table, stocks) }

Coordinations = { ((treemap, select), (table, select)) }

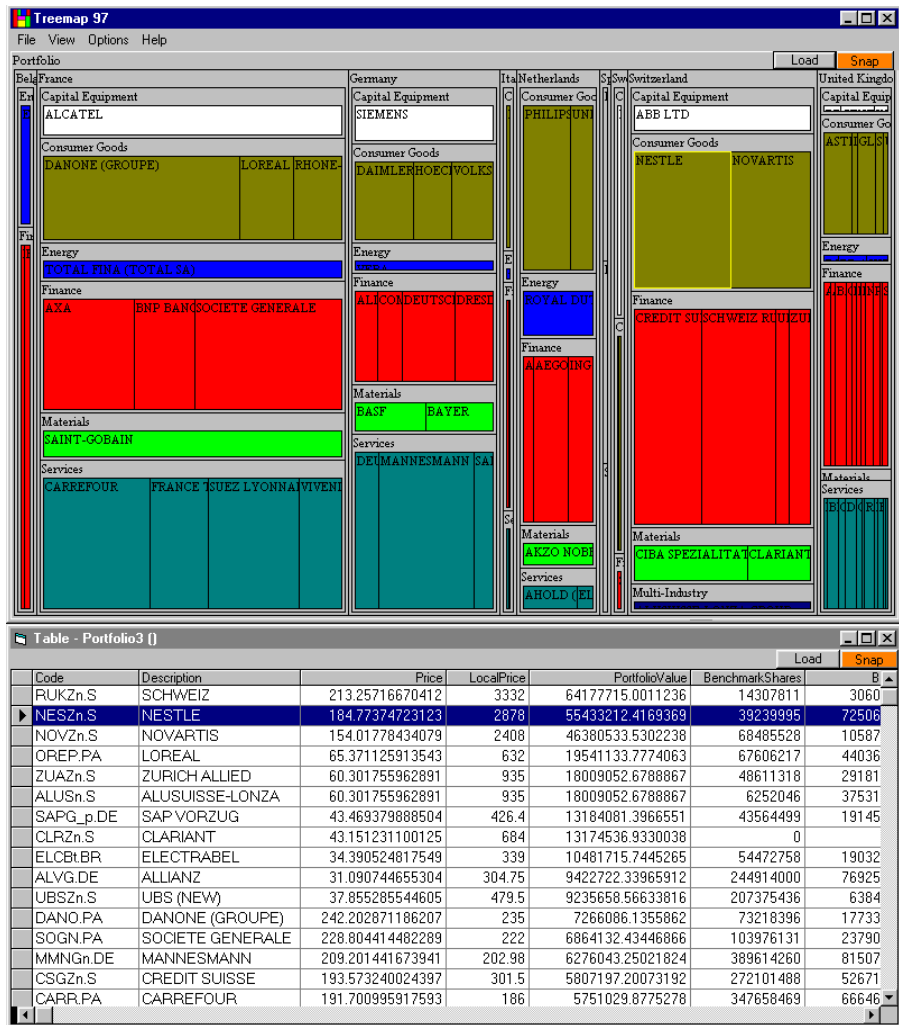


Figure A.8: Stock market portfolio scenario

A.9 Visible Human Images

As described in Chapter 3, Snap could also be used in medical and scientific domains to relate physical structures in images to other types of information. The mockup in Figure A.9 demonstrates the concept. While this example has not been implemented, it can be done with Snap. For example, html image maps in IE have been used with Snap. An image map of the U.S. was used prior to the use of ArcView

(Figure A.10). Another approach might be to use a volume visualization tool that supports the selection of structural objects.

The Snap specification for this interface would be:

```

Visualizations = { (volumeviz, structures), (outliner, structures) }
Coordinations = { ((volumeviz, select), (outliner, select)) }

```

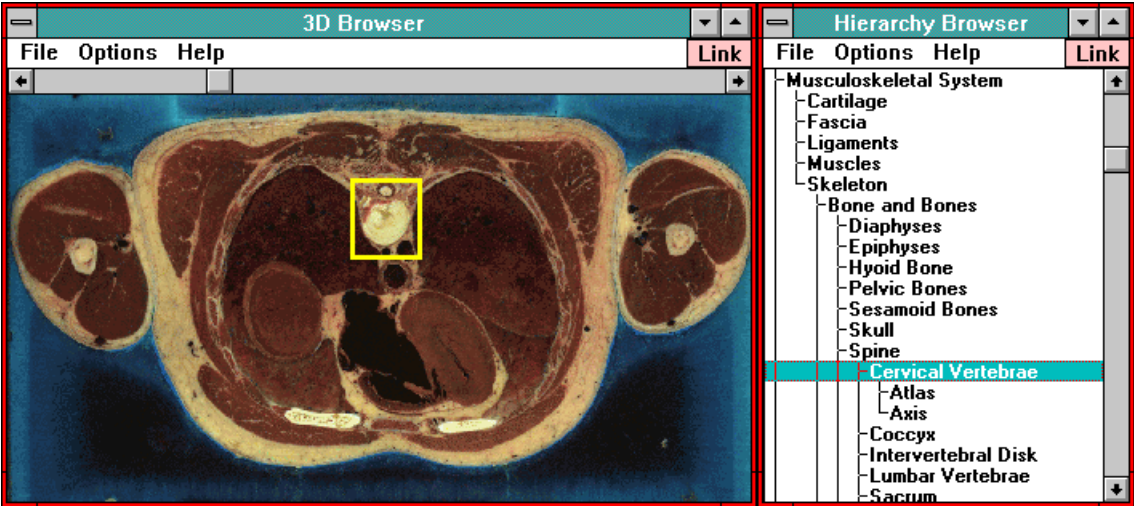


Figure A.9: Visible Human images scenario

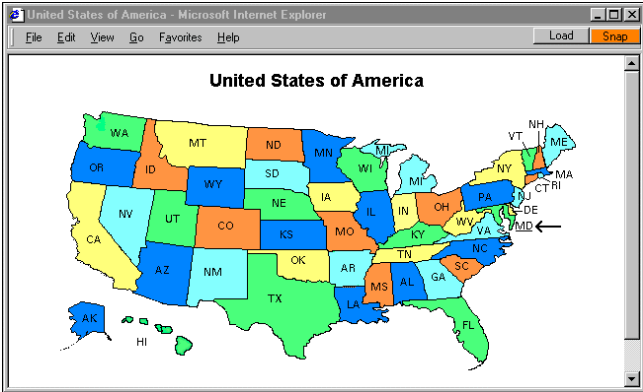


Figure A.10: Image map in IE

A.10 Summary

These scenarios also demonstrate the serious need that Snap fulfills. Without the use of Snap, scenarios such as the web logs example simply could not be readily accomplished. They would require significant custom programming, or the difficult and tedious use of uncoordinated displays.

These examples demonstrate how Snap has already been highly applicable and useful in many projects. It has been useful to both researchers and practitioners, and has already had an impact at several organizations including HCIL, the Census Bureau, Spotfire and WestLaw.