

# DESIGNING INTERACTIVE VISUALIZATIONS FOR FIRST-TIME NOVICE USERS

Sujatha Krishnamoorthy

*Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of*

***Master of Science***

*In*

***Computer Science***

## **Advisory Committee:**

Dr. Chris North (Chair)

Dr. Scott McCrickard

Dr. Manuel Pérez-Quñones

November 9<sup>th</sup>, 2005  
Blacksburg, Virginia

**Keywords:** information visualization, interface design, novice users, learnability

# **DESIGNING INTERACTIVE VISUALIZATIONS FOR FIRST-TIME NOVICE USERS**

Sujatha Krishnamoorthy

Committee Chair: Dr. Christopher L. North

## **Abstract**

Information visualization tools provide visual representations of data (commonly known as visualizations), textual representations of data and interactive operations on both these representations. It is possible, in theory, to make use of only the textual representations in order to detect trends and patterns in data. However, it would be extremely laborious and ineffective and it defeats the purpose of a visualization tool. Novices have had the problem of relying on text and failing to successfully detect trends in data because it was too laborious. We want users to effectively use visual representations to detect trends in data.

Information visualization tools have been shown to be successful with experts. But can novice users using a visualization tool for the first-time, adopt visualization-based strategies to finding trends and patterns in data?

This thesis derives a framework of learnable elements in an interactive coordinated-view visualization tool. This framework provides an outline of prerequisites to be learned in order to effectively use visualizations. That is, the new aspects of visualization tools must be mastered so that novices can use the tool effectively.

Three interface design principles are derived to make these elements learnable to novice users:

- The data-first approach - Provide a prominent overview of all available data, as opposed to showing only visualizations. This helps understand the data structure – this may be essential knowledge in being able to navigate to required data attributes.
- The “less visualization, more explanation” approach –Show more explanations of visualizations at the expense of being able to pack in more visualizations. Explanations help identify how data is mapped onto visual marks, a crucial step in understanding visualizations.
- The “predetermined task-based coordinations” approach – Provide separate predetermined sets of coordinated visualizations that help achieve different tasks. This is different from techniques that simply present all visualization types and expect users to choose coordinations according to tasks.

Two versions of Datamaps visualization tool for Census data were tested. Both were equivalent in functionality and the kinds of visualizations offered. But the new version was specifically designed based on the three design principles. A usability study showed that the version that was implemented according to the three design principles successfully led novices to effectively use visualization-based strategies to detect trends and patterns in data.

# Acknowledgements

Dad and Mom, the best moments of my life were whenever you truly believed in me – those are the moments that keep me going. Thanks go to my sisters Geetha and Savitha, and brothers-in-law Madhusudan and Venkateswaran, for their love and support. Thank you, my grandmothers, Anantha-lakshmi Ammal and Muthu-lakshmi Nilakantan for your love and blessings.

Many thanks go to Dr. North for being a wonderful mentor. His “Information Visualization” course was the first time I experienced something that is both technical and wildly creative. He was contagiously cheerful and enthusiastic. I thank Dr. Pérez-Quñones for his guidance and advice. His insightful questions and observations helped me figure out how to present my work and write up this thesis. He is an inspiration, especially in the field of computer science education – something which has always sparked interest in me. Many thanks go to Dr. McCrickard for being a great teacher. Attending his “Notification Systems Research Group” talks gave me my first exposure to the process of creating theories and definitions. His “Usability Engineering” course was an eye opener. To me, excellent design no longer meant “a sudden stroke of inexplicable genius” – it became something which could be achieved through systematic consideration of tradeoffs and priorities.

During the course of my master’s program I have been fortunate to interact with some great teachers. Every conversation that Dr. Tatar and I have had – from experimental design to balancing life’s priorities – is etched in my mind. She has helped me see that every problem can be viewed from several perspectives, and each one gives you a different solution. I will always remember Dr. Shelli Fowler for her encouragement – she has taught me a lot about teaching as a profession. I am fortunate to have worked as assistants to Mr. Harrison who is one of those rare individuals who can effortlessly mix art and technology and Dr. Fox who is my ideal for productivity – he amazes me with how much he can get done in a single day.

I am indebted to fellow Datamaps researchers Qing Li, Daniel Keathley, Tyler Hennessy, Matthew Phillips, Johnny Sam-Rajkumar, Pradyut Bafna, Will Lee, Andrew and Umur Yilmaz. Each of their efforts in Datamaps proved to be extremely valuable in my research. I am grateful to the U.S Census Department for funding the research on Datamaps.

I appreciate “the lab” for their suggestions and encouragement – Purvi, Beth and Bob. I am grateful to my good friends at VT who have helped me at times of need – Jamika, Vikram, Karthik, Kiran, Likhitha, Glenn Fink and his wife Sherril Fink. Thanks to Tess, Rachel, Carol, Jessie, Renee, Netta, “Dee Dee” and Ginger at the Computer Science office – they keep this department going and were always there for me if I needed help.

I would like to express my warm regards to my high school friends for always believing in me and encouraging me – Sharmishta, Shankar, Ramanathan, Mahadevan, Karthik, Vijay, Ramesh, Krishna, Arvind, Ranganath, Yogesh, Charulekha, Praveen, Vasiullah and others. A little chat with them always cheers me up. To this day, these are the people that make me feel on top of the world. Thanks go to my roommate Priyadarshini Ramachandran, for being like a sister. An enormous “Thank you” goes to Johnny for being the greatest buddy in the whole world.

I remember with love, my teachers at the University of Madras, where I did my Bachelors degree in Computer Science and Engineering- Professors Rajendran, Maragatham, Padmasani, Muneeshwari, Sharmila and others made me discover good things about myself and helped me develop those skills. Mrs. Saraswathi (my history teacher) and Mrs. Jayalakshmi (my mathematics teacher) at middle school remain close to my heart, always. I miss them every time I think of my childhood.

Finally, thanks to The Great Power of the Universe, for giving me all these wonderful people and learning experiences.

# Table of Contents

<b>List of Tables.....</b>	<b>x</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 The Problem.....	1
1.2 Datamaps: The tool studied in this work.....	2
1.3 The Novice User .....	5
1.4 The Power of Visualization: A Scenario.....	5
1.5 Difficulties for novices to Information Visualization .....	8
1.6 Research question .....	9
1.7 Significance of this work .....	11
1.8 Content – this document in a nutshell.....	13
1.9 Key to reading this document .....	14
<b>2 Motivation .....</b>	<b>15</b>
2.1 Root Concept .....	15
2.2 Design of Original Datamaps.....	15
2.3 Design rationale for Original Datamaps .....	23
2.4 Usability study on Original Datamaps .....	24
2.4.1 Performance of subjects – qualitative results .....	25
2.4.2 Performance of participants – quantitative results.....	30
2.5 Summary .....	32
<b>3 Related Work .....</b>	<b>33</b>
3.1 Visualization tools for novices.....	33
3.1.1 Dynamic HomeFinder: real estate visualization.....	33
3.1.2 ActiveGraph and Envision: document collection visualization.....	34
3.1.3 Mail-Sleuth: email visualization.....	36
3.2 Discussion of visualization tools for novices.....	37
3.3 Designing for novices: techniques and previous research .....	37
3.3.1 Integrated instruction technique .....	38
3.3.2 Exploration technique.....	38
3.3.3 Animation technique.....	39
3.3.4 Sticky notes.....	39
3.3.5 Complexity in levels or layers .....	41
3.4 Discussion on the techniques and previous research .....	41
<b>4 Learnable Visualization Elements Framework.....</b>	<b>43</b>
4.1 Cognitive Artifacts.....	43
4.2 Effective distributed cognition systems .....	44

4.3	A user plus visualization tool system.....	45
4.4	Need for a framework of learnable elements in a Visualization Tool.....	45
4.5	Taxonomies for Information Visualization.....	46
4.6	Using taxonomies to build a framework of learnable elements in a Visualization tool.....	49
4.7	The Learnable Visualization Elements Framework .....	50
4.7.1	Data.....	50
4.7.2	Visualizations .....	51
4.7.3	Coordinations.....	53
4.7.4	Overall interface level controls.....	53
4.8	Learnable Visualization Elements Framework Definitions .....	54
4.9	Knowledge in the Head for target users .....	55
4.10	Difference between a novice and expert .....	55
4.11	Summary .....	57
<b>5</b>	<b>Design Hypotheses .....</b>	<b>58</b>
5.1	The “Data-first” approach and Hypothesis 1 .....	58
5.2	The “Less Visualization for More explanation” approach and Hypothesis 2 .....	59
5.3	The “Predetermined Task Based Coordinations” approach and Hypothesis 3 .....	60
<b>6</b>	<b>Design of Datamaps .....</b>	<b>61</b>
6.1	Description of final design.....	61
6.2	Design iterations and rationale.....	67
6.2.1	First Design Iteration .....	67
6.2.2	Second Design Iteration.....	72
6.2.3	Final Design Iteration and Rationale .....	75
6.3	Summary .....	77
<b>7</b>	<b>Usability Study .....</b>	<b>79</b>
7.1	Objectives of the study.....	79
7.2	Differences between the two versions of Datamaps .....	79
7.3	Experiment Procedure.....	84
7.4	The two visualization-based strategies possible in the experiment.....	89
7.5	Rationale for recommending visualization-based strategies .....	90
7.6	Performance of participants – quantitative results .....	90
7.7	Performance of participants – qualitative results .....	93
7.8	Discussion of usefulness of new design approaches.....	94
7.8.1	The data-first approach .....	94
7.8.2	“Less visualization for more explanation approach” .....	98
7.8.3	Predetermined Configurations approach .....	101
7.9	Tradeoffs involved in designing for novices.....	102
<b>8</b>	<b>Conclusions and Future Work.....</b>	<b>103</b>

8.1	Conclusions.....	103
8.2	Potential implications.....	105
8.3	Future work.....	105
<b>9</b>	<b>References.....</b>	<b>108</b>
<b>10</b>	<b>Appendix – A: Demographics Questionnaire.....</b>	<b>110</b>
<b>11</b>	<b>Appendix B – Screening test .....</b>	<b>113</b>
<b>12</b>	<b>Appendix C: Main Experiment Task List.....</b>	<b>115</b>
<b>13</b>	<b>Appendix D: Exit Questionnaire .....</b>	<b>118</b>
<b>14</b>	<b>Appendix E: Informed Consent Form .....</b>	<b>121</b>
	<b>Informed Consent Form .....</b>	<b>121</b>
	<b>I. THE PURPOSE OF THIS RESEARCH/PROJECT .....</b>	<b>121</b>
	<b>II. PROCEDURES .....</b>	<b>121</b>
<b>III.</b>	<b>RISKS.....</b>	<b>121</b>
<b>IV.</b>	<b>BENEFITS OF THIS PROJECT .....</b>	<b>121</b>
<b>V.</b>	<b>EXTENT OF ANONYMITY AND CONFIDENTIALITY .....</b>	<b>121</b>
<b>VI.</b>	<b>COMPENSATION .....</b>	<b>122</b>
<b>VII.</b>	<b>FREEDOM TO WITHDRAW .....</b>	<b>122</b>
<b>VIII.</b>	<b>APPROVAL OF RESEARCH .....</b>	<b>122</b>
<b>IX.</b>	<b>SUBJECT'S RESPONSIBILITIES AND PERMISSION .....</b>	<b>122</b>

# List of Figures

Figure 1-1: An example of using the visual representation .....	2
Figure 1-2: A screenshot of old Datamaps .....	3
Figure 1-3: A single visualization provides insight.....	6
Figure 1-4: Coordinations between visualizations provide insights .....	7
Figure 1-5: Coordination between map view and scatter-plot view:.....	11
Figure 2-1: Original Datamaps screenshot. ....	16
Figure 2-2: Original Datamaps: histograms with dynamic query slider widgets.....	17
Figure 2-3: Original Datamaps: Map colored by an attribute.....	17
Figure 2-4: Scatter-plot in Original Datamaps .....	18
Figure 2-5: Original Datamaps: Detail table showing numerical figures for states selected on the map .....	19
Figure 2-6: Original Datamaps: Detail table showing numerical figures .....	20
Figure 2-7: Sticky note help in Old Datamaps .....	21
Figure 2-8: Original Datamaps: Coordinations .....	22
Figure 2-9: Table with default attributes at start up.....	26
Figure 2-10: Laborious scrolling through tables of data.....	27
Figure 2-11: Configuring the map component to show color codes.....	28
Figure 2-12: Configuring the scatter-plot .....	30
Figure 3-1: Dynamic Homefinder(Williamson 1992) .....	34
Figure 3-2: ActiveGraph Visualization of search result sets(Marks 2005).....	35
Figure 3-3: Envision for digital libraries (Lakkaraju 2004).....	36
Figure 3-4: Mail-Sleuth for email visualization(Eklund 2004). ....	37
Figure 3-5: Sticky notes for integrated initial guidance.....	40
Figure 4-1: Visualizations are cognitive artifacts .....	43
Figure 4-2: Pfizner’s taxonomy for Information Visualization .....	47
Figure 6-1: New Datamaps at start-up (Single map configuration).....	61
Figure 6-2: Map with legend window open.....	62



Figure 6-3: Table shows values for all selected states for current attribute.....	63
Figure 6-4: Bar histograms .....	63
Figure 6-5: New Datamaps with “Map + Histograms” configuration .....	64
Figure 6-6: The dynamic legend explaining visual syntax .....	65
Figure 6-7: New Datamaps showing “Multiple maps” configuration .....	66
Figure 6-8: An initial design using the Data-first approach .....	68
Figure 6-9: New layout led users to organize actions-paths from data to visualization .....	71
Figure 6-10: Right-click menu interaction to create visualizations for attributes.....	72
Figure 6-11: Choosing a map on the menu loads a map.....	73
Figure 6-12: Size and tiling are adjusted to fit more visualizations.....	73
Figure 7-1: Configuring the map to show color in old Datamaps .....	80
Figure 7-2: Same map acts as dynamic map and choropleth map in old Datamaps .....	80
Figure 7-3: Choropleth map and Dynamic map on different tabs in new Datamaps.....	81
Figure 7-4: Pie charts to test graphical reading ability of participants .....	85
Figure 7-5: Left-to-right action path in configuration 1, New Datamaps.....	96
Figure 7-6: Left-to-right action path in configuration 2, New Datamaps .....	97
Figure 7-7: Labels help understand visual syntax .....	99
Figure 7-8: Legends help understand visual syntax.....	100
Figure 7-9: Selection-Deselection confusion without the legend in Old Datamaps .....	100

# List of Tables

Table 1-1: Differences between everyday visualizations and coordinated-view visualization tools.....	8
Table 1-2: Chapters and details .....	14
Table 2-1 Original Datamaps: Definition of coordinations .....	22
Table 2-2: Tasks for the usability study on Original Datamaps .....	25
Table 2-3: Participant performance on Original Datamaps .....	31
Table 4-1: Knowledge in the World and Head must complement each other .....	44
Table 4-2: A framework of learnable elements in a visualization tool.....	54
Table 6-1: Example working of dynamic legend.....	65
Table 6-2: Summary of design iterations.....	67
Table 6-3: Claims for design iteration-1.....	69
Table 6-4: Claims for design iteration-2.....	74
Table 6-5: Learnable Elements in Datamaps and design features that convey them.....	77
Table 7-1: Differences between the two tested versions of Datamaps .....	83
Table 7-2: Group A demographics .....	86
Table 7-3: Group B demographics .....	87
Table 7-4: Task list for usability study on new Datamaps.....	88
Table 7-5: Strategies of Group A (old version) & Group B (new version) participants.....	91
Table 7-6: Number of people using visual strategies .....	92
Table 7-7: Number of times visualization-based strategies were used.....	92

# 1 Introduction

## 1.1 The Problem

Information visualization tools are front-end interfaces to relational databases at the back-end. They offer:

- Visual representations of data,
- Textual representations of data and
- Interactive mechanisms that operate on both kinds of representations.

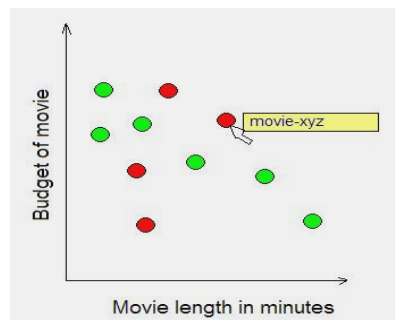
Visualization tools are commonly designed with the intention that the visual representations must provide overviews and the text must provide details about specific items that may look interesting from the visual overview.

Now, it is possible to find trends and patterns in data using only the textual representations and interactive mechanisms that operate on the textual representations. But this means that a person would have to laboriously read through detailed descriptions of all data objects in order to detect trends in data. This is extremely time consuming and ineffective. It does not leverage the power of human perception in detecting trends and patterns from visual representations.

This kind of problem has been observed with novice users. Novices resort to reading textual representations instead of visual representations to detect trends in data. This may be attributed to the fact that text is familiar to them. So they may be more comfortable with the textual representation than with visual representations. But then, not using the visual representation would imply that the whole purpose of the visualization tool is defeated. **We want novice users to effectively use the visual representations and**

**interactive mechanisms that operate on the visual representations to find trends and patterns in data.**

Consider this scenario. A user understands that all red circles are English movies and that the positions of the dots indicate the budget and movie length (this is shown in Figure 1-1). Then the user wants to know the name of one circle because it is both high-budget and long-length. She moves the mouse over that circle to know the movie name. She is still interested and clicks on it to pop up textual details about the movie. This is effective usage of both visual and textual representation.



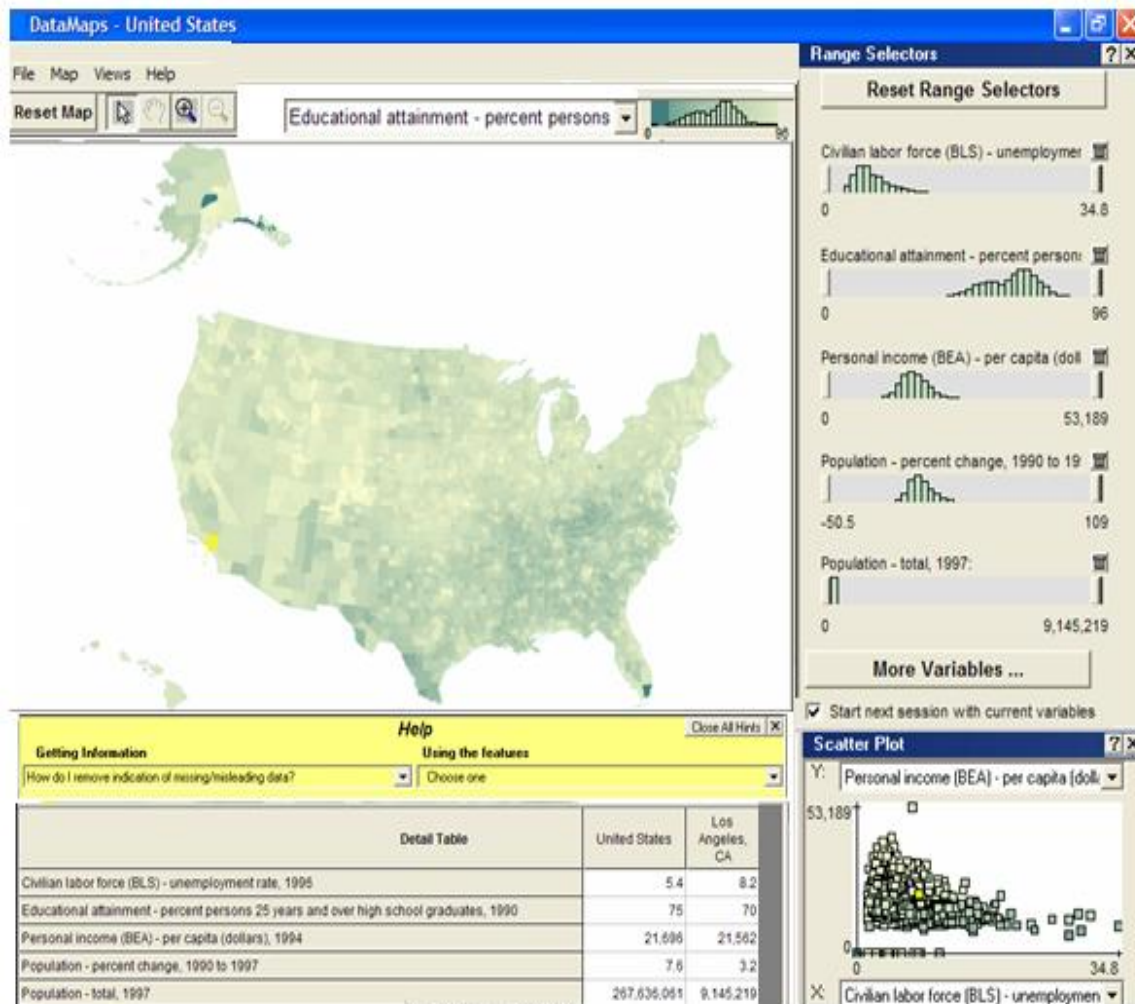
**Figure 1-1: An example of using the visual representation**

Information Visualization tools have shown to be successful with experts. But **can first-time novice users of interactive visualizations use visual representations to explore data?** Given that they have no prior experience with interactive visualizations and little or no familiarity with the data set itself, how can we lead novices to adopt visualization-based strategies to gain insights about data?

## **1.2 Datamaps: The tool studied in this work**

This work tries to answer these questions by taking the case of Datamaps. Datamaps is a software tool intended to be an online interactive visualization of Census

data for visitors to the US Census Bureau website. There are around 8000 attributes such as “population 1997”, “unemployment rate, 1996” collected at the state level and at the county level. Datamaps acts as a visual interface to this large database. Figure 1-2 shows a snapshot of Datamaps.



**Figure 1-2: A screenshot of old Datamaps**

**Datamaps** fared well with information visualization experts but not with **novices**. The discrepancy in their ability to effectively adopt visualization-based strategies was the motivation behind this work. This thesis derives and studies the effect

of design principles that lead first-time novice users to adopt visualization-based strategies.

The target audience for Datamaps is loosely defined as “the general population needing Census data on US states and counties”. There might be users who are “just looking around”, users who require specific data for making business decisions, researchers collecting data, students doing geography projects, census bureau officials working with data, divorce lawyers looking at marriage statistics by state and so on. Business decision makers, politicians, policy makers and school teachers are more examples. Datamaps may be characterized in the following ways:

- It is a relational data visualization tool (for Census Data on US states)
- It is multi dimensional (Nearly 8000 dimensions, since each state has around 8000 data attributes like “number of crimes known 1988”, “unemployment rate 1990”, etc. Note: attributes are also known as dimensions in Information Visualization parlance)
- It provides multiple interactive coordinated visualizations (map, plot, histograms and a table. Figure 1-1 shows a snapshot of Datamaps with these views.)
- It is intended to be part of a web-site and will potentially be used in short one-time exploratory sessions. So users must be able to use visualization-based strategies in the first usage session without the need for tutorials or large amounts of help documentation.

## 1.3 The Novice User

Since Datamaps is general purpose software available on a website, it is likely to be visited by occasional or one-time users. A user who comes to consult Datamaps on the Census web site is very likely to be comfortable with graphical user interfaces and simple graph interpretation.

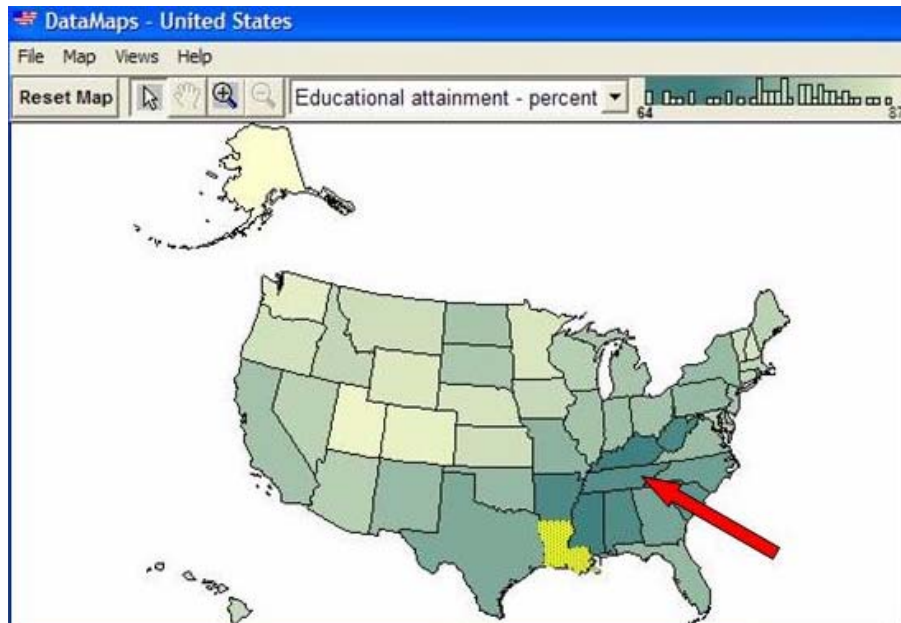
An informal survey revealed that bar-charts and pie-charts are the most commonly encountered and easily understood visualizations for those who have no prior exposure to interactive information visualizations. That does indicate that although user skills in interpreting visualizations may be limited, basic familiarity using visual representations of data widely exists. **Novices are therefore defined as users who:**

- **Don't have expertise in the underlying data** that is being visualized. That is, they have not collected, experimented with or organized the data. They merely come to consult the data.
- **Don't have prior exposure to information visualization** tools.
- Do have **familiarity with graphical user** interfaces. That is, they can perform actions such as pointing, clicking, double clicking, right clicking, typing alphanumeric characters and they are familiar with elements such as windows, icons and menus on the interface.
- Do have **graph reading ability** for simple static charts: bar charts, pie charts etc.

## 1.4 The Power of Visualization: A Scenario

An assistant to policy makers at a university is gathering information on income and educational attainment levels in order to make a report. She does not know much

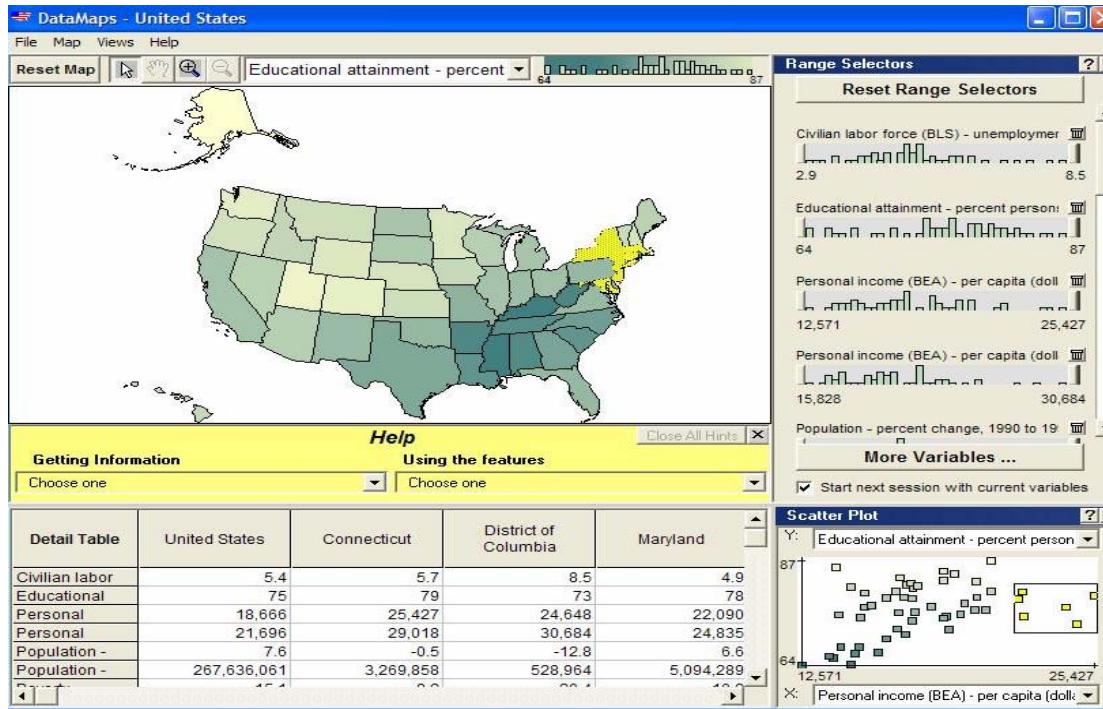
about trends in income and education levels. She consults Datamaps. She looks at a map in which states are colored by income: darker the color, lower the income. Figure 1-3 shows the color-scale map. At a glance she locates a cluster of states in the south with low income. This turns out to be a good observation for her report.



**Figure 1-3: A single visualization provides insight: Low income states are present in a cluster in the south**

Next, she looks at a scatter-plot, the y-axis showing educational attainment levels and the x-axis showing income. She chooses a set of dots on the scatter-plot that have high income. They are highlighted. Correspondingly, some states on the north-east are highlighted on the map. Figure 1-4 illustrates the highlighted dots on the plot and corresponding states on the map. “The highest income states all seem to be clustered in the northeast” she observes. Observations such as this are possible because there are coordinating views: in this case, a map and a scatter-plot.





**Figure 1-4: Coordinations between visualizations provide insights: High income and medium education level states are on the north-east**

Information **visualization leverages the power of the human perception**. Human perceptive abilities are remarkably good at scanning, recognizing and recalling images, detecting changes in size, color, shape, movement, or texture in a visual presentation (Shneiderman 1996). Understanding visual representation of data is a parallel process – one can look at vast amounts of data all at once.

Plaisant points out that **visualization tools “help answer questions that you never knew you had”** (Plaisant 2004). The above scenarios illustrate this. Prior to looking at the visualizations, the user did not have ideas about trends. The visualization prompted queries in the user’s mind. In the sense conveyed by Norman in (Norman 1991), a visualization tool is in fact a cognitive artifact which changes the user task from textual-queries to immediate perception of data and interactive querying.

## 1.5 Difficulties for novices to Information Visualization

Although information visualization is powerful, novices may face difficulties with information visualization tools. Information visualization tools might be “too much to take in”. Wurman speaks of “information anxiety” and “information overload” experienced by users (Wurman 1989).

Simple visualizations are encountered everyday on television, in presentations, books, reports, newspapers and websites. Some examples are bar graphs, pie charts, weather maps etc. Even though novice users may have encountered them in daily life, and even though they may possess graphical reading ability, they might still face difficulties with multiple-view visualization tools. This is because there are some aspects about multiple-view information visualization tools that make them more complex than everyday visualizations. Table 1-1 summarizes some key differences between them.

Simple everyday visualizations	Visualizations in a multiple-view visualization tool
<b>Not Configurable</b>	<b>Configurable:</b> the same visualization type can be configured to display different sets of data. For example, a scatter-plot tool that is displaying “birth rate versus death rate” can be configured to show “marriage rate versus divorce rate”. This is new to novice users.
<b>Not Interactive</b>	<b>Interactive:</b> Operations such as selection and navigation may be performed, configuration changes may be made, parameters may be changed etc.
<b>Employ very commonly used mappings:</b> Typically, visualizations encountered in everyday life are simple and commonly used types such as colored pie charts, labeled bar charts etc. Mappings are as simple as color for categories, height for numerical values etc.	<b>Employ innovative mappings:</b> Information visualizations may employ mappings as unique as shape or texture of a visual mark to represent categorical data. There are novel representations of data. For example, a tree-map which slices and dices a rectangular space to show hierarchical structures.
<b>Most probably isolated:</b> Visualizations are most probably embedded within text and not packed in with other visualizations.	<b>Most probably accompanied by many more visualizations:</b> As the name “multiple-view” indicates, there are probably multiple visualizations open at the same time, causing increased screen complexity.

**Table 1-1: Differences between everyday visualizations and coordinated-view visualization tools**

The novel aspects of information visualization tools might cause obstacles in a novice's ability to adopt visualization-based strategies if the interface is not designed well enough to overcome those obstacles.

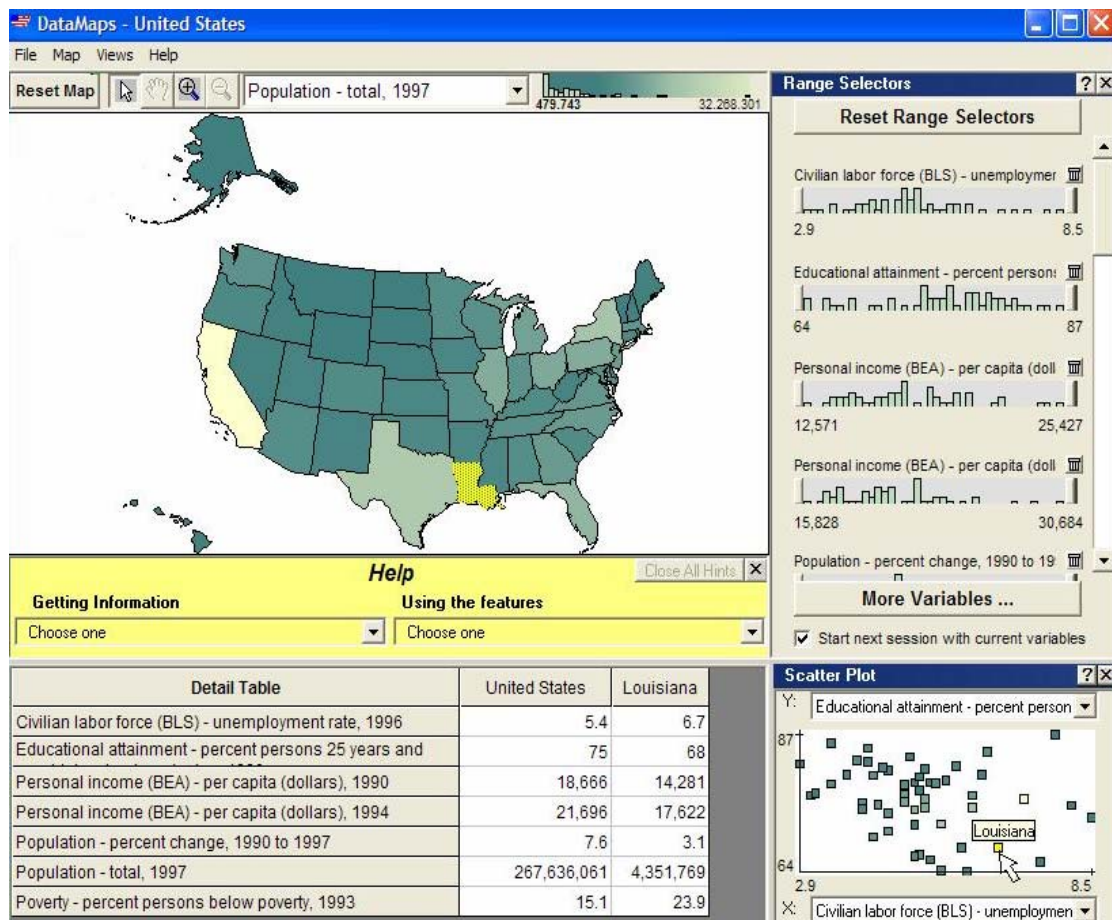
## 1.6 Research question

On the one hand there is power: the human perception system is so powerful that it is advantageous to use information visualizations to provide insights. On the other hand, there are difficulties: since many aspects about information visualization tools would be new to novice users. So they may not understand the tool properly and may not be able to employ visualization-based strategies to detect patterns. **We aim to bring the power and benefits of information visualization to novice users while minimizing the difficulties.**

So the research question is “**Can interface design enable first-time novice users of a coordinated-view visualization tool to employ visualization-based strategies to gain insights from data? If so, how?**” The important terms are clarified below:

- **Interface design:** There are several methods by which graphical user interfaces can explain themselves to their users: help documents, animated tutorials, manuals etc. These are all extrinsic in several ways – they are typically non-contextual (requiring adaptation from generic instructions to current situation); they are spatially separate from the main interface (appearing in new windows) and they are temporal interruptions to normal flow of actions in the interface (had the user known what to do). The approach here is to examine interface design principles which intrinsically increase understandability of the interface, while minimizing the use of extrinsic aids.

- **First-time novice user:** “First time” simply refers to the fact that the user is exposed to the tool for the first time. There could be a first-time user who is an expert in the data – they study or analyze the data, or have collected the data. We are not dealing with these kinds of first time users. There could be a first-time user who has seen other information visualization tools – we are not dealing with them either. We are specifically dealing with first-time novice users. We are trying to enable novices to employ visualization-based strategies in the very first usage session, because it may after be one-time use, and they may never need it again.
- **Coordinated-view visualization tool:** A coordinated view visualization tool is one that shows different visualizations or views at the same time, and has them coordinate with each other. As in North’s Snap-Together model (North 2000), a coordination is defined as an action-invocation pair. An action in one visualization (or view) is coupled to an action in another visualization (or view). The two views which are coordinated in this fashion are said to be “coordinated views”. For example in Datamaps, clicking on a state on a map highlights the state and also the dot on the scatter-plot that corresponds to that state. A selection action on the map invokes a selection action on the plot. Figure 1-5 shows this.



**Figure 1-5: Coordination between map view and scatter-plot view: Clicking on Louisiana on the map has highlighted it both on the map and on the plot.**

## 1.7 Significance of this work

Good lessons from perception-level research are widely available to designers of information visualization. For example, Cleveland (Cleveland 1984), Tufte (Tufte 1990), Christ (Cleveland 1984; Cleveland 1985) and Mackinlay (Mackinlay 1986) present findings about design of graphical encodings for perception. The Gestalt principles of perception are often-cited authoritative findings in designing for perception (Koffa 1935). However, barring a few medical exceptions, perceptive abilities are the same for all human beings. **Perception level research does not address the distinction between**

**experts and novices** since they all perceive in the same way. This research is focused on how design can influence higher level of human information processing, such as thinking, evaluating and adopting strategies on the interface.

Graphical encoding guidelines may lead designers to focus on individual visualizations. However, **this research focuses on the design at the interface-level** – what design principles enable novices to adopt visualization-based strategies.

There are a considerable number of visualization-rich data exploration interfaces for experts in domains such as bioinformatics, computer network monitoring, geography, meteorology etc. But advanced visualization-rich interfaces are rare for novices. **If human perception is powerful, why not take advantage of that and bring advanced visualizations in the case of novices as well?** With that in mind, this work focuses on bringing advanced visualization-rich interfaces to novices.

This work and more similar research may make important **contributions to designing visual interfaces for everyday interfaces like search engines, email inboxes etc. that must reach out to novice users.** Search engines, even small ones that are interfaces to small databases, accept form based textual input and deliver textual results. Form based inputs hide data. By contrast, a visualization would show trends in the entire document set. This might be especially useful in digital library search interfaces. Email inboxes almost always show tabulated textual information: “From”, “subject”, “date”, “time” etc. Visualizations instead of tables might help see structure of email exchanges, trends and patterns in the kinds of email received etc.

## **1.8 Content – this document in a nutshell**

A census database visualization tool called Datamaps fared well with experts, but not with “off the street” novice users. Novices resorted to text-based strategies in the interface to solve tasks during a usability study. They resorted to reading tables of values rather than interpreting visualizations and interacting with them. This slowed them down to the extent that they could not proceed with given tasks. Results of this study motivated this work.

In this work, a framework of “learnable visualization elements” was formed. This framework helps outline those aspects of a coordinated-view visualization that might look new to a first-time novice user. Based on this framework, three hypotheses were generated about design principles that should lead novices to adopt visualization-based strategies. Scenario Based Design approach was used in generating new information layouts and interaction design for Datamaps. Several iterations of design ensued until a final new version of Datamaps was reached.

A think-aloud usability evaluation test was used to compare original Datamaps and the new one with all three design hypotheses implemented. The number of times visualization-based strategies were adopted (instead of “table reading” strategies) was used as a metric for success. It is shown that novice users are indeed able to adopt visualization-based strategies designed with these three design principles.

## 1.9 Key to reading this document

Table 1-2 gives an overview of the rest of the chapters in this document.

Chapter	Details
2) Motivation	<ul style="list-style-type: none"><li>• Discusses a usability test with Datamaps visualization tool – it was successful with experts, but fared poorly with novices</li></ul>
3) Related Work	<ul style="list-style-type: none"><li>• Reviews literature on design techniques for novices</li></ul>
4) Learnable Visualization Elements framework	<ul style="list-style-type: none"><li>• Considers “Visualization tool plus novice user” as a distributed cognition system.</li><li>• Derives a framework of learnable elements in a visualization tool (Knowledge in the World)</li><li>• Discusses what typical novices know while they arrive as first-time users (Knowledge in the Head)</li></ul>
5) Derivation of Design Hypotheses	<ul style="list-style-type: none"><li>• Derives three design hypotheses based on the LVE Framework and the needs of novices</li></ul>
6) Design of New Datamaps	<ul style="list-style-type: none"><li>• Discusses new design and implementation of Datamaps Visualization tool based on the hypotheses</li><li>• Discusses the iterations required to arrive at final design</li></ul>
7) Usability Study	<ul style="list-style-type: none"><li>• Describes experiment conducted to study two versions of Datamaps: one designed according to the three design hypotheses and one designed without any of the three approaches.</li><li>• Describes successes of the new version based on a count of number of times users were able to adopt visual thinking strategies in each interface</li></ul>
8) Conclusions and Future work	<ul style="list-style-type: none"><li>• Discusses the contributions of this work, potential impacts and outlines areas that need further research.</li></ul>

**Table 1-2: Chapters and details**



## 2 Motivation

### 2.1 Root Concept

**The vision behind Datamaps is to provide visual representations of Census data to users. The rationale is that visualizations leverage human perceptive abilities** into obtaining insights about large quantities of data. Datamaps is meant to be an online tool on the Census Bureau website available to anyone accessing the site. Therefore it is desirable that novices be able to adopt visualization-based strategies in the very first usage session.

The product was designed in keeping with principles in Information Visualization, and was successful with expert users who were used to data analysis. But usability tests revealed that “off the street” novice users were not using visualization-based strategies, but relying only on their ability to read tables. The motivation for this work came from usability test results on Datamaps. The design of original Datamaps and the usability study conducted by the Census Department are discussed in this chapter.

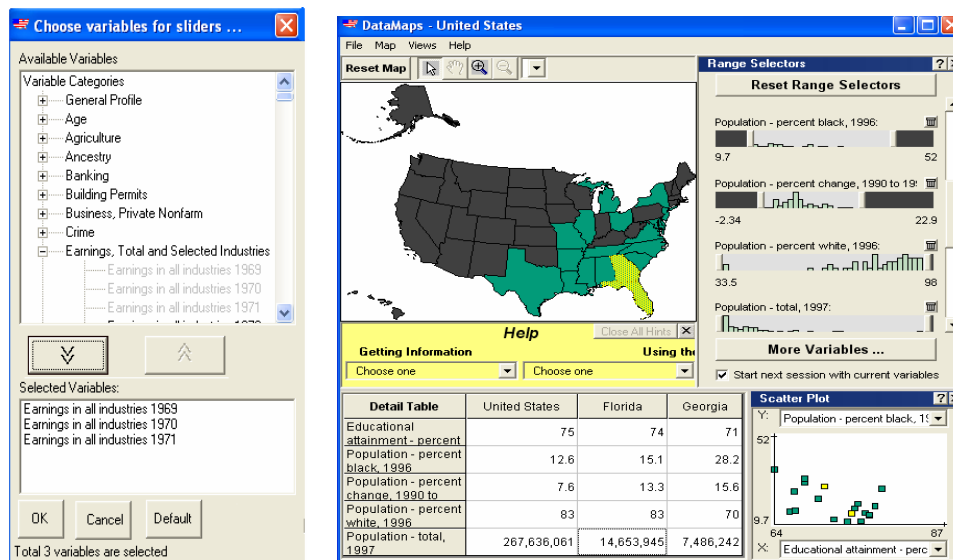
### 2.2 Design of Original Datamaps

Datamaps is a tool for visual analysis of census data on the United States. Data has been collected for approximately 8000 attributes. “Attributes” are items such as “total population” “percentage white population” “percentage black population” and so on. These attributes are grouped together by categories such as “Age”, “Agriculture”, “Banking”, “Crime” etc. Figure 2-1 shows Datamaps with map view, a histogram view, a table, and a scatter-plot view. When the application is opened, five attributes are selected by default:

- 1) Civilian Labor force unemployment rate 1996
- 2) Educational attainment, percent persons 25 years& above, high school graduates 1996
- 3) Population total 1997
- 4) Population percent change 1990 to 1997
- 5) Personal Income (BEA) per capita income (dollars) 1994

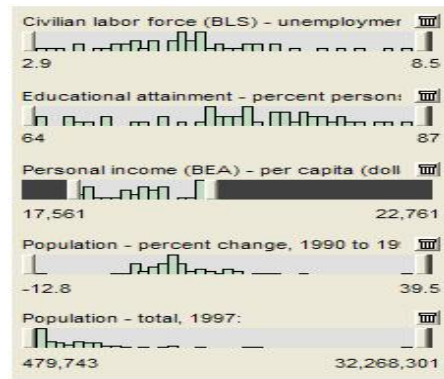
Five corresponding histograms show the frequency distributions for these attributes.

If the user needs to visualize attributes other than these five, he/she has to click the “more variables” button. A tree-view of the 8000 attributes (as seen in Figure 2-1) would appear on a separate window and allow the user to select new attributes to view.



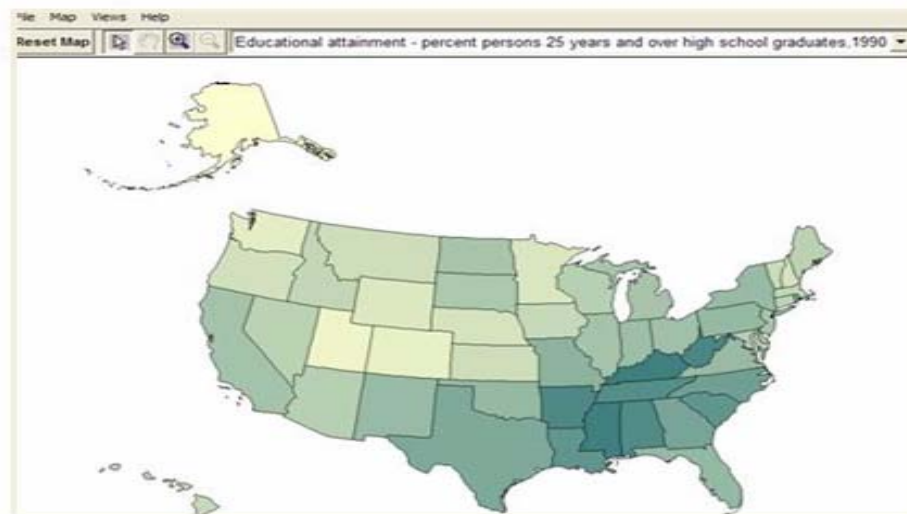
**Figure 2-1: Original Datamaps screenshot. Main window with map, table, histograms with dynamic query widget and scatter-plot. “Choose variables” window shows categorized list of Census attributes.**

The histogram shows a frequency distribution. The height of each bar stands for the number of regions occurring in predetermined class intervals. Several histograms are simultaneously visible. Figure 2-2 shows the histograms section of Datamaps, with the five default attributes that are loaded at start-up:



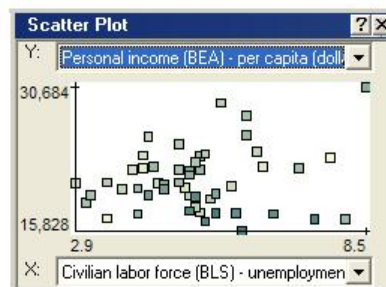
**Figure 2-2: Original Datamaps: histograms with dynamic query slider widgets**

The map view employs color gradation to show attribute values as a function of geographic areas. Darker regions imply lower numerical values. Only one map is viewable at a time. Figure 2-3 shows a map colored by the “Educational attainment” attribute. Alaska is seen to be very pale (greater educational attainment) and a cluster of states including Alabama, Arkansas, Mississippi, Kentucky, Tennessee and West Virginia are darker in color (lesser educational attainment in these states).



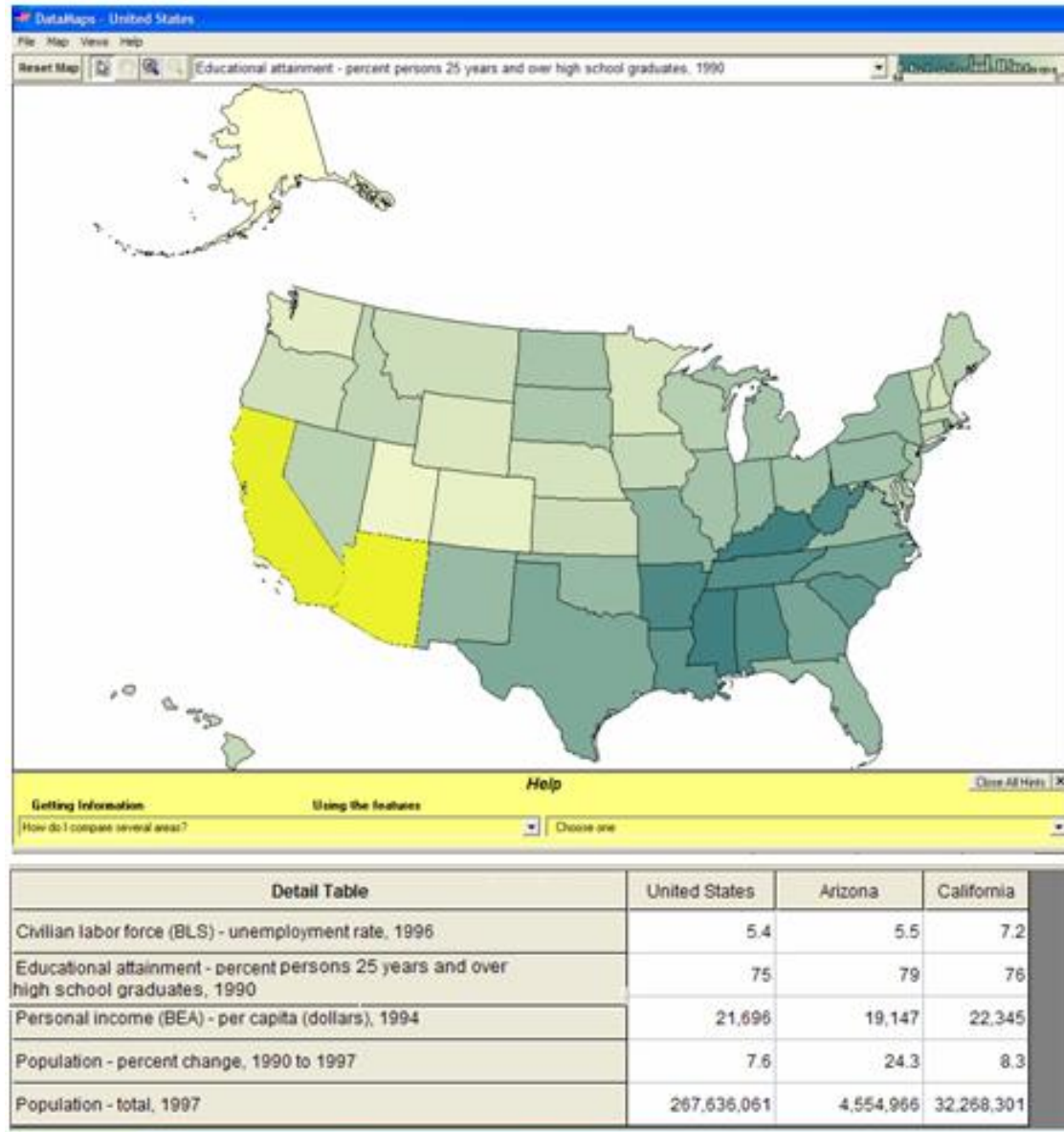
**Figure 2-3: Original Datamaps: Map colored by an attribute.**

The scatter plot aids in assessing the relationship between two attributes. Each dot represents a region (a state, if operating in “states mode” or a “county” if operating in “counties mode”). The scatter-plot is configured with two default attributes. To view the relationship between any two attributes (other than the two default attributes), the X-axis attribute and Y-axis attribute must be chosen from the drop down list. The dots are colored by whatever attribute colors the map (irrespective of which attributes are on the X- and Y- axes). Figure 2-4 shows a scatter-plot of states by “Civilian Labor Force uemployment rate” versus “Personal income per capita”. The dots are colored by “educational attainment”, since the map was colored by “educational attainment” at the time.



**Figure 2-4: Scatter-plot in Original Datamaps**

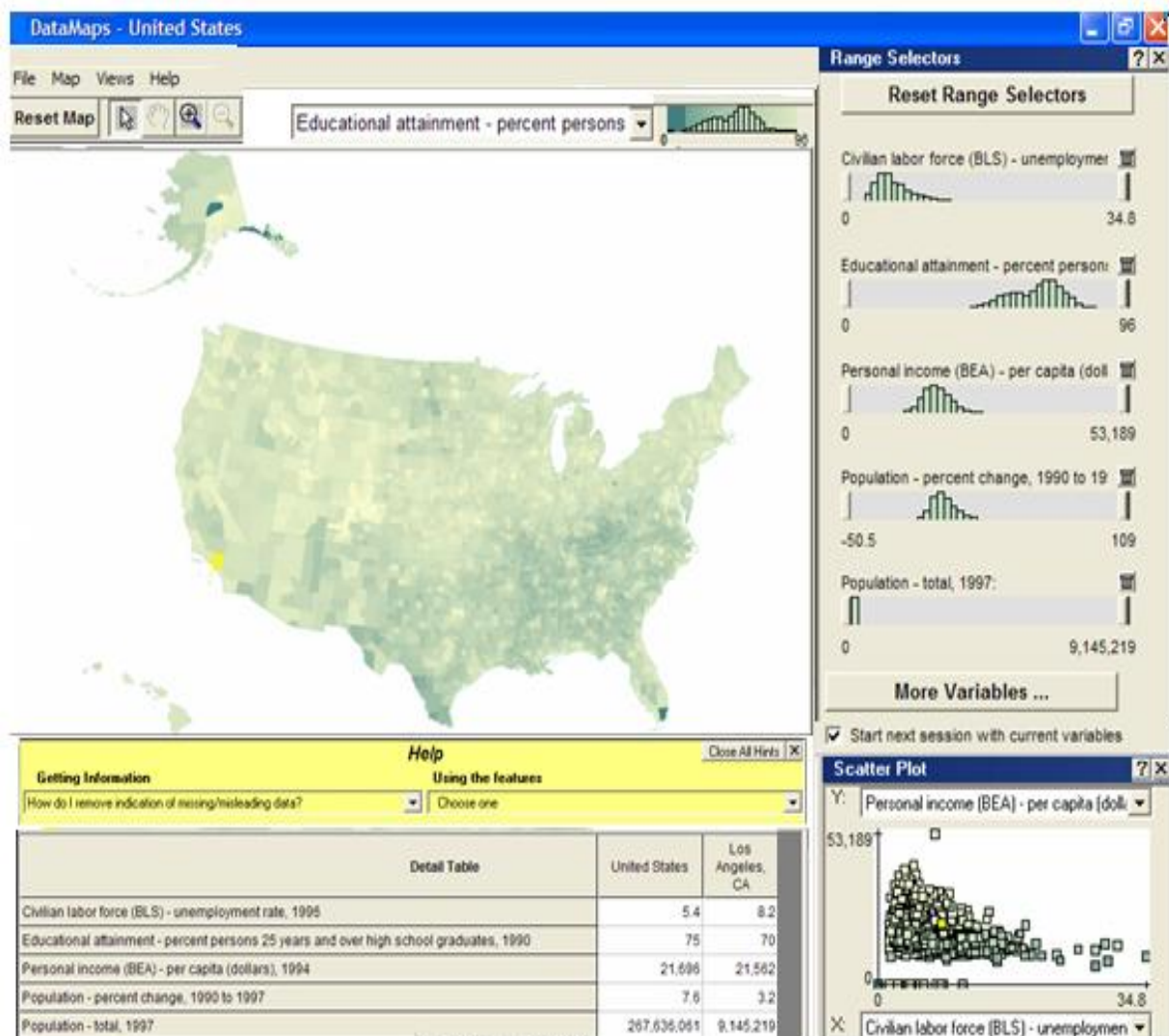
The table below the map is activated upon clicking on a region on the map or a dot on the scatter-plot. It loads data values for that state (or county) and presents the textual figures in tabulated format. Figure 2-5 shows California and Arizona selected (highlighted) on the map, and loaded on the table. The table displays data for all the currently loaded attributes.



**Figure 2-5: Original Datamaps: Detail table showing numerical figures for states selected on the map**

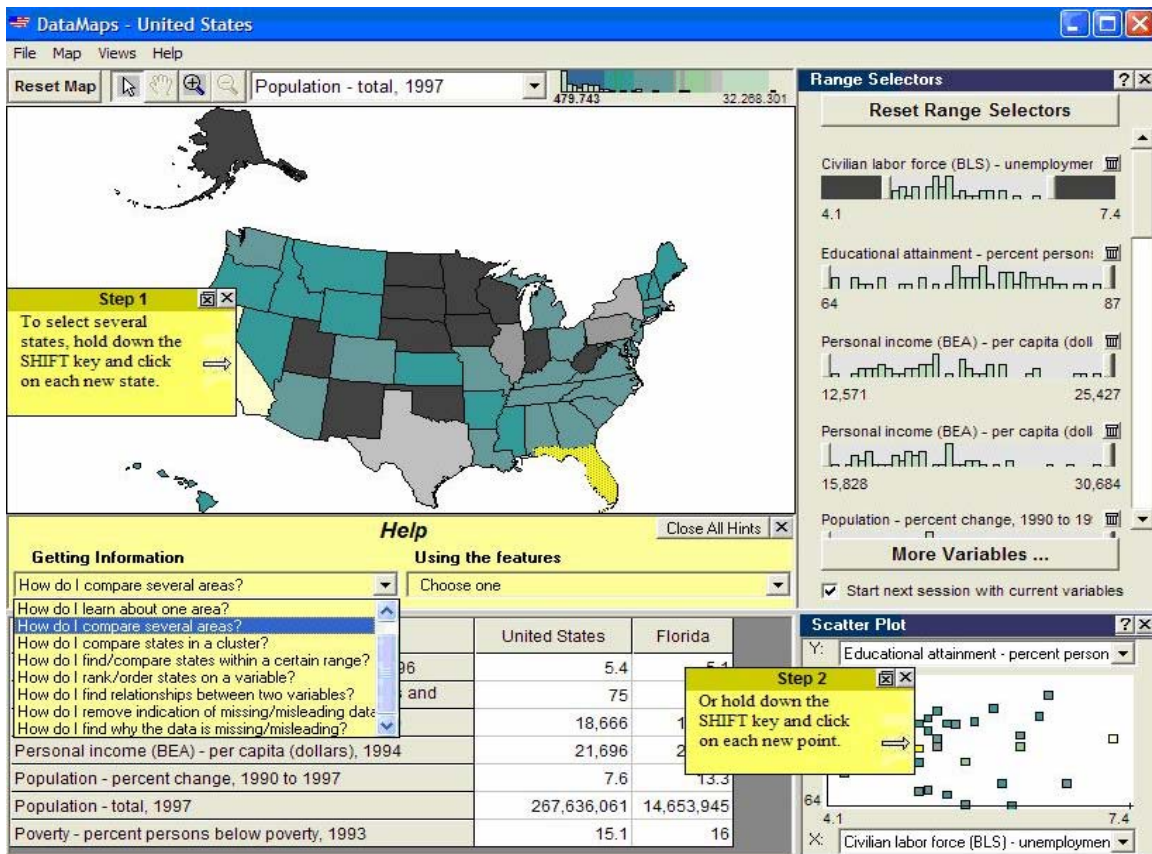
Datamaps operates in “states mode”, “all US counties mode” or “counties of a particular state” mode. In the “states” mode, the entire interface including the map, table, histograms and plot is tuned to show Census data collected at the state level. Datamaps can be set to operate in “counties” mode. In this case, all the visualizations show counties level data: the map colors individual counties, scatter-plot shows dots for counties and

histogram shows bars that stand for number of counties. Figure 2-6 shows Datamaps in the counties mode. The county of Los Angeles, California is highlighted on the map and scatter-plot. The table displays all data for that county.



**Figure 2-6: Original Datamaps: Detail table showing numerical figures**





**Figure 2-7: Sticky note help in Old Datamaps**

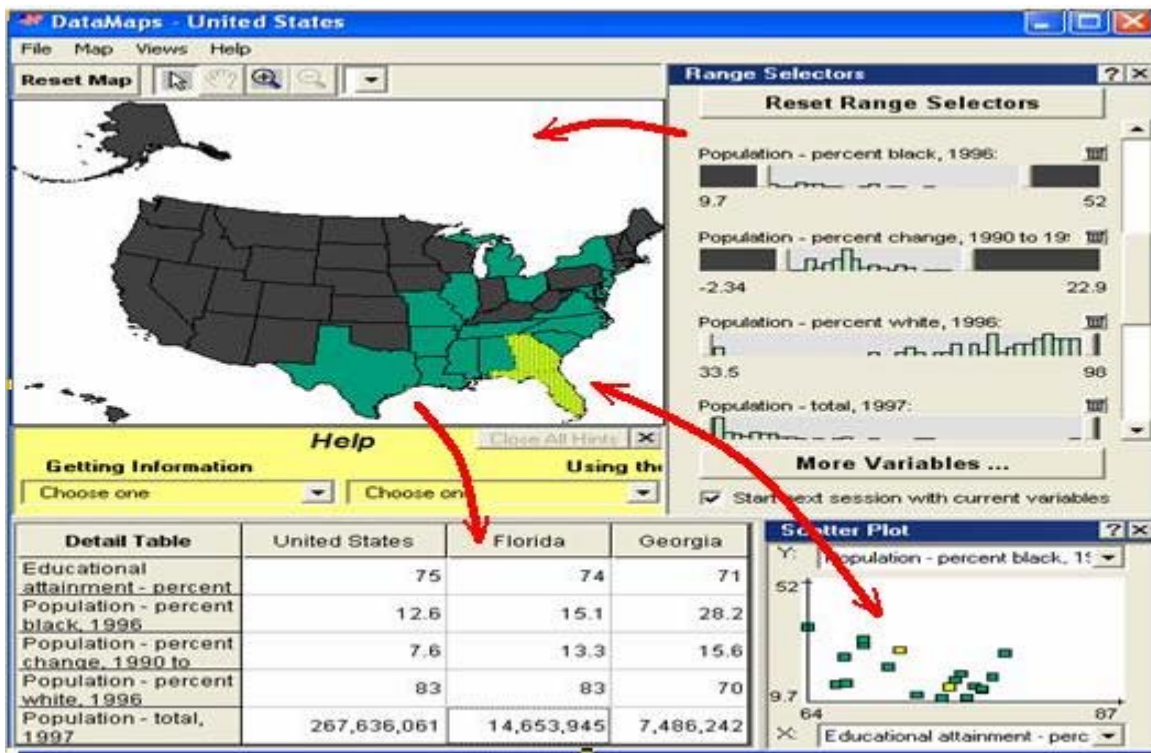
Datamaps comes with a special kind of help – sticky notes. Figure 2-7 illustrates sticky note help. In the help section, two drop-down boxes are available: “Getting Information” and “How do I use the features”. Each drop-down box has a set of help questions. When a help question is chosen, the related sticky notes pop up on the interface. These are step-wise instructions to get a task done. If a step relates to a particular portion of the interface, then that sticky note is placed near that part of the interface. In figure 2-7, the sticky notes describe how to compare several areas. The yellow sticky note near the map says “To select several states, hold down the SHIFT key and click on each new state”. The sticky note near the plot says: “Or hold down the SHIFT key and click on each new point”.

Table 2-1 gives a summary of the coordinations between the visualizations.

Coordination	Description
(Map, Selection), (Table, Load)	Clicking on a region on the map highlights that region on the map and loads a column of data for the corresponding region in the table
(Map, Selection), (Scatter-plot, Selection)	Clicking on a state/county on the map highlights that region on the map and also highlights the corresponding dot on the scatter plot
(Scatter-plot, Selection), (Table, Load)	Clicking on a dot on the scatter-plot highlights that dot on the scatter-plot and loads a column of data for the corresponding region in the table
(Scatter plot, Selection), (Map, Selection)	Clicking on a dot on the scatter-plot highlights that dot on the scatter-plot and also highlights the corresponding region on the map.
(Histogram, De-selection), (Scatter-plot, De-selection)	Moving the sliders filters out bars in the histogram, and the corresponding states in the scatter-plot disappear.
(Histogram, De-selection), (Map, De-selection)	Moving the sliders filters out bars in the histogram and the corresponding states in the map are de-selected (darkened out)

**Table 2-1Original Datamaps: Definition of coordinations**

Figure 2-8 illustrates the coordinations:



**Figure 2-8: Original Datamaps: Coordinations**



## 2.3 Design rationale for Original Datamaps

Overall interface design was based on Shneiderman's Information Visualization Mantra "Overview first, zoom and filter, then details on demand" (Shneiderman 1996). The map component would act as the overview. That is, the map coordinates with all other views. Datamaps is intended to be used by a wide variety of users. Average users will best recognize states on a map, rather than states represented as dots. So the map is used as the primary overview. The interface has been designed, keeping the following action path in mind:

- Understand all available visualization types that are available by looking at visualizations of default attributes;
- Select desired attribute(s);
- Configure desired visualization type to show desired attribute(s);
- Look at visualizations of desired attribute(s) and obtain insights;
- Interact with visualizations to make queries as desired

Several software tools have empty screen at start up; the user is expected to look at menus or buttons and activate something to get started. This leaves first-time novice users clueless about the interface. Datamaps tries to avoid the "empty screen" problem. It provides all available visualizations at start-up – map, scatter-plot and histograms. They are loaded with default attributes.

Users may not know where in the interface one can find all 8000 Census data attributes. Having the default set of attributes at start-up makes them get used to working

with attributes. The “More attributes...” option is placed near the default set. This is expected to “ease” users into seeing the complete set of attributes.

Datamaps was partly inspired by the design of Spotfire, which is now a commercial visualization tool. Spotfire (Ahlberg 1996) is a dynamic query based general purpose visualization tool. It gives an overview of several hundreds data items in one view. One of the basic visual marks used is circles. They are color coded for categorical data– user may specify which colors go for which categories. Dynamic query filters help quickly filter out spots based on desired value ranges. Dots that do not fall in the desired range dynamically disappear from view, as the range is dynamically varied.

## **2.4 Usability study on Original Datamaps**

The Census department conducted usability studies on the version of Datamaps that was described in the previous section. Four novice users were given 10 tasks and asked to think aloud while trying to complete each task. They were asked to “play around with the interface” during a pre-task session and describe their first impressions. The sessions were video-taped and mouse movements on screen were recorded. All of users were quick to remark that they would click on a state to view information about it, within only a few seconds of playing with the Data-maps interface. Table 2-2 lists the tasks that the novice users were required to perform.

S.N	Question
1	What was the population of California in 1997?
2	Compare the 1994 income level of people in 2 states not next to each other. Which had greater per capita income?
3	Of the western states, which state had the lowest unemployment rate in 1996?
4	Name the states where 20% to 22% of the population in 1990 had high school degrees.
5	Which two counties in Nevada had the highest population percent change from 1990 to 1997?
6	How many families in Minnesota lived below the poverty line in 1989?
7	You are thinking of moving to a new state. You want to live in a state that has low poverty level, high income level, and low unemployment rate. Which state or states best fit these criteria?
8	You are interested in graphing the relation between high school graduation and one's personal income across the states. How would you do it?
9 a	Which county in the USA had the highest population in 1997?
9 b	You would like the map to show number of persons below poverty level for each county. How would you do that?
10	You no longer want to play around with one data item. How will you remove it from your view?

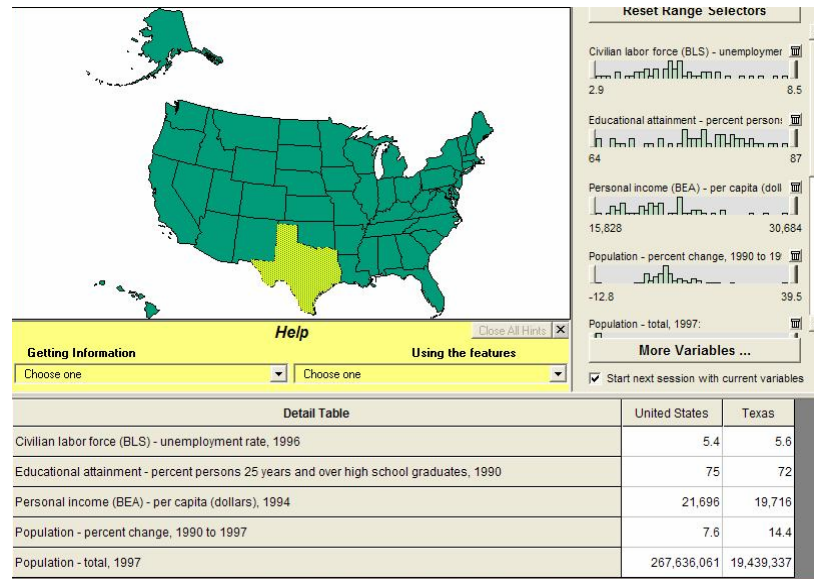
**Table 2-2: Tasks for the usability study on Original Datamaps**

### 2.4.1 Performance of subjects – qualitative results

The video recordings of the four users showed that user goals were loose chunks of information needs: “what” “when” “where” and “how much/many”. They sought cues in the interface that would probably contain answers to those chunks. For example, let us consider the question: “What was the total population of California in 1997?” The identifiable chunks in this question are:

- “What was” – the “how much” or the data value (required answer)
- “the population” – the “what” or the attribute
- “California” – the “where” or the object of interest or the tuple
- “1997” – the “when” or the time of data collection

The users would look for “chunks” of their information need in several portions of the display, and if found, interact with those chunks. Initial tasks were easy to do by just trying to look at matching words on the interface. They turned out to be easy because they involved the five attributes that were present by default in the table. The table with the default attributes is shown in figure 2-9.

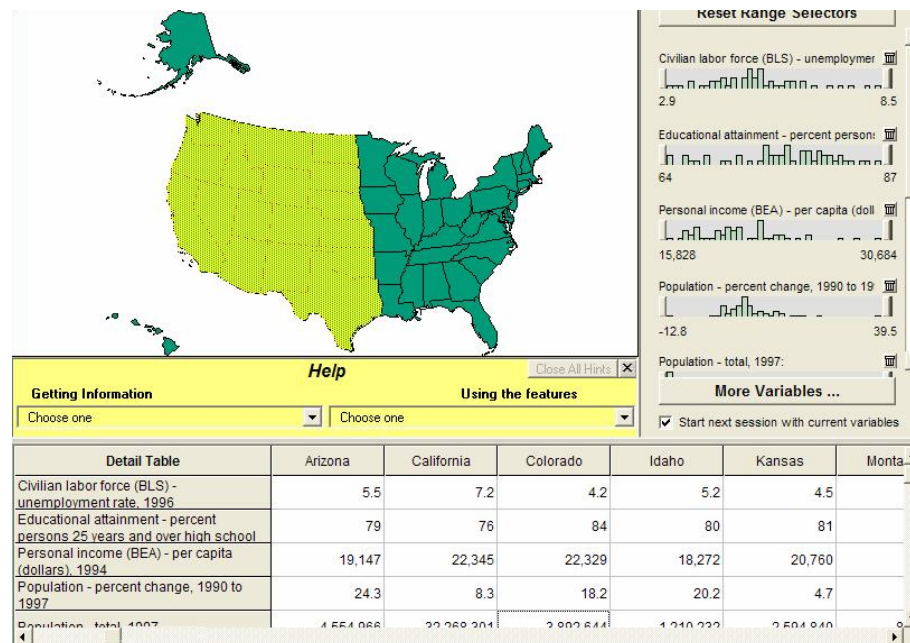


**Figure 2-9: Table with default attributes at start up**

Different kinds of usability problems are described below:

- Unclear action paths:** The initial tasks involved a small number of states and default attributes. However, as tasks grew more complex, involving all states, or involving attributes that were not present by default on the table, users were totally unclear how to proceed. The time spent on various portions of the interface, the order and reasons for visiting those portions were indicative of disorganized, broken action plans. They could not figure out procedures and action paths in the interface.

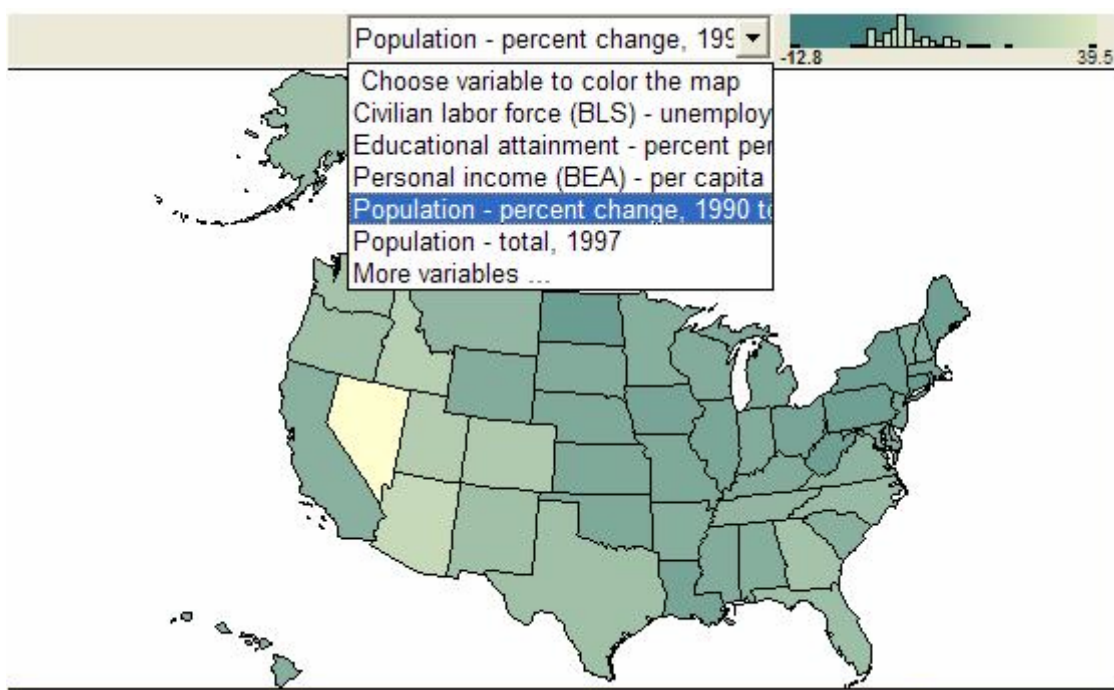
- Latching on to inefficient strategies (tables for everything):** The coordination between the map and the table was very easy to learn. This can be attributed to the fact that the map is probably the most prominent and familiar entity on the screen and hence invites clicking. This also has immediate feedback: clicking on a state on the map loads values for that state in the table. Users latched onto this strategy. When faced with a question such as “which western state in the USA had the highest population percent change from 1990 to 1997”, users selected all western states and scrolled the table in search of the highest value. Figure 2-10 shows a populated table and gives an idea of the laboriousness of scrolling and reading values.



**Figure 2-10: Laborious scrolling through tables of data**

In order to locate the maximum, they could only follow the tedious algorithmic procedure of scrolling the table, because they were armed with the “click map and read table” strategy. It required excessive mental book-keeping; and they gave

up. Simply selecting the given attribute to color the map would have helped locate the maximum valued region by color tone, but they could not figure out this method. Figure 2-11 shows the drop down box from which the attribute “Population percent change 1990-1997” could have been chosen. This would have colored the map as shown, and it would have been easy to tell that the answer was Nevada, from the colors and the legend on top of the map. It is interesting to note that only one user could discover the use of color, and it was by accident, not by looking at the legend on the top right of the color-coded map.



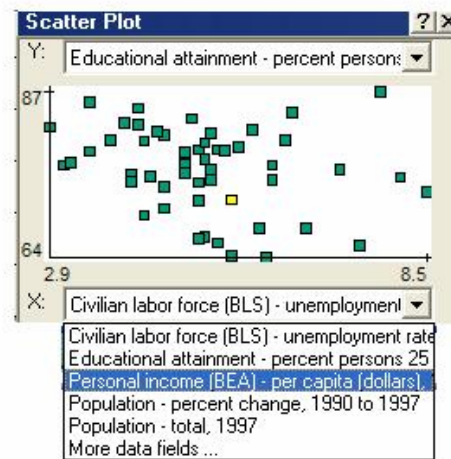
**Figure 2-11: Configuring the map component to show color codes**

- **Understanding interaction mechanisms or widgets:** The slider widgets resemble buttons. There are visual cues to suggest “drag-ability”. It is the dragging action that helps select a sub range and filters out (deselects) states not in that sub-range. It operates similar to Dynamic Queries in Home Finder

visualization (Williamson 1992). The left slider button must be dragged to set the lower bound and the right to set the upper bound. Two of the users did not even attempt to use them. One participant clicked on each button but couldn't figure out that they must be dragged in order to be used. One moved the slider during the pre-task, but it she never used it to solve any of the given tasks.

- **Understanding visual syntax:** One user serendipitously discovered that the map is showing colors and he began to suspect that the colors mean something. Instead of looking at the legend, he tested his hypothesis by clicking on light colored and dark colored states and reading off their values. Only this user was able to use color (which is the visual syntax for map visualization) in answering the complex tasks. Two users did not even consider looking at the bar graphs. Two users wrongly interpreted the histograms to be bar graphs and wondered what the bars meant. One user remarked that the bars were “Scary”. He wondered if each bar stood for a state. Another user wanted something to pop up when she clicked on the bars. Only one person used the scatter-plot correctly. It was not possible to tell whether syntax was a problem, because the presence of this visualization component was hardly acknowledged by three users.
- **Skipping the configuring stage of visualization components:** The scatter-plot visualization does not visualize the latest selected attributes, unless it is configured to do so. Selecting the attributes from the tree view does not automatically make the plot show the latest selected attribute. The newly selected attributes would be loaded only in the drop down boxes against the scatter-plot. They have to be chosen again, to make the plot show those two attributes. One

user selected two attributes from the attribute list while answering the “graph the relationship between two attribute” question. But this user could not figure out how to configure the plot to show the two newly selected attributes. Figure 2-12 shows how the x-axis and y-axis must be configured to show the required attributes. Three users never configured the map to become colored by an attribute. One user accidentally, rather than intentionally colored the map, and later found out meanings of the color with educated guesses.



**Figure 2-12: Configuring the scatter-plot**

#### **2.4.2 Performance of participants – quantitative results**

Table 2-3 summarizes the strategies used by the participants. The dark gray color indicates that the corresponding tasks were approached by using a visualization-based strategy. The lighter gray cells show tasks which were achieved by reading tables or tasks in which users were completely unclear about how to proceed. It is notable that only six of the forty tasks were done by using visualization-based strategies. Five of these were done by one user alone. Users were unclear about several tasks (note cells marked “unclear”) and primarily resorted to using the map as a navigation tool for reading the table (note cells marked “by table”).



	TASK	User 1	User 2	User 3	User 4
<b>1</b>	What was the population of California in 1997?	<b>By table</b>	<b>By table</b>	<b>By table</b>	<b>By table</b>
<b>2</b>	Compare the 1994 income level of people in 2 states not next to each other. Which had greater per capita income?	<b>By table</b>	<b>By table</b>	<b>By table</b>	<b>By table</b>
<b>3</b>	Of the western states, which state had the lowest unemployment rate in 1996?	<b>By table</b>	<b>By table</b>	<b>By table</b>	<b>By color</b>
<b>4</b>	Name the states where 20% to 22% of the population in 1990 had high school degrees.	<b>Unclear:</b> "Should I use the bars? The plot? The table?"	<b>Unclear:</b> "No data attribute looks useful"	<b>By table</b>	<b>Unclear:</b> "How/where to specify the year to be 1990?"
<b>5</b>	Which two counties in Texas had the highest population percent change from 1990 to 1997?	<b>By table</b>	<b>Unclear:</b> "All counties of USA" mode scatter-plot confuses him.	<b>By table</b>	<b>By color</b>
<b>6</b>	How many families in Minnesota lived below the poverty line in 1989?	<b>By table</b>	<b>By table</b>	<b>By table</b>	<b>By table</b>
<b>7</b>	You are thinking of moving to a new state. You want to live in a state that has low poverty level, high income level, and low unemployment rate. Which state or states best fit these criteria?	<b>Unclear:</b> "Too tedious to click all states!" Quit	<b>By table</b> Quit after a while	<b>By table</b> Quit after a while	<b>By table</b> Reduced the question to 4 states
<b>8</b>	You are interested in graphing the relation between high school graduation and one's personal income across the states. How would you do it?	<b>Unclear:</b> Tried to graph it from the table & quit	<b>Unclear:</b> "I don't understand scatter-plot"	<b>Unclear:</b> Could not configure axes of plot	<b>By plot</b>
<b>9 a</b>	Which county in the USA had the highest population in 1997?	<b>Unclear:</b> considered sliders, reading table	<b>Unclear:</b> considered clicking bars, reading table	<b>By plot</b> Confused about x- y- axes	<b>By color</b>
<b>9 b</b>	You would like the map to show number of persons below poverty level for each county. How would you do that?	<b>Unclear:</b> Unclear about meaning of "below poverty". Quit	<b>Unclear:</b> "That would be too much info!" Quit.	<b>Unclear:</b> Quit	<b>By color:</b>

By table = approached the problem by reading table; By plot = approached the problem by interpreting the scatter-plot; By color = approached the problem by interpreting colors on map; Unclear = had various misunderstandings in interface or question; no clear strategy; no solution reached

**Table 2-3: Participant performance on Original Datamaps**

## 2.5 Summary

The usability test has revealed interesting results with novices. They mostly either resorted to tables or didn't know what to do at all. Although Original Datamaps was designed based on well known guidelines and previous similar tools, it did not fare well with novices. **Novices did not adopt visualization-based strategies in the original Datamaps tool.** This creates the research question: "How can we design coordinated-view visualization tools that lead first-time novice users to adopt visualization-based strategies?" The next chapter reviews literature on designing interfaces for novices.

## 3 Related Work

### 3.1 Visualization tools for novices

Interactive visualization tools in domains like bioinformatics are different from visualization interfaces for domains like digital libraries. The former kind would be most likely used by “experts”: people who have collected and experimented with the data. They would be very familiar with the data set as a whole. The latter kind will be used by novices. In the digital library example, patrons will not be experts on the whole document collection. Only a librarian would be such an expert. It is the latter kind of visualizations that is the focus of this work. **Datamaps is intended to be used by novice users – not by experts on the data set such as those who collect or compile Census data.** A description of similar visualization tools for novices follows.

#### 3.1.1 Dynamic HomeFinder: real estate visualization

Dynamic HomeFinder (Williamson 1992) implements dynamic queries. This is probably the closest visualization to Datamaps since it involves spatial information and dynamic querying. Dots represent houses that are for sale. Each house has attributes such as “number of bedrooms”, “cost” etc. Dynamic query sliders are used to set the value range criteria, for example: “number of bedrooms must be greater than 2 and less than 6”. Figure 3-1 shows a snapshot of HomeFinder. According to an experiment by Williamson and Shneiderman, “Novices can learn to use the [dynamic query] system quickly with both query formulation and results displayed in task domain” (Williamson 1992). It is encouraging that this interface has worked for novices. **However the differences between Dynamic HomeFinder and Datamaps may be the source of complexity in Datamaps.** Some key differences are:

- **Homefinder deals with houses; whereas the data objects in Datamaps are either states or counties.** More general knowledge is required to work with geographical regions than houses.
- **The attributes in Homefinder are short and simple, whereas in Datamaps they are long.** Consider the simplicity of “Cost” attribute from Homefinder in contrast with an attribute like “Commercial Banks and Saving Institutions (FDIC insured) – number of offices, 1997” in Datamaps.
- **Homefinder has very few attributes, Datamaps has nearly 8000.** The attributes are grouped into related categories. Navigating them and finding required attributes is much more difficult.

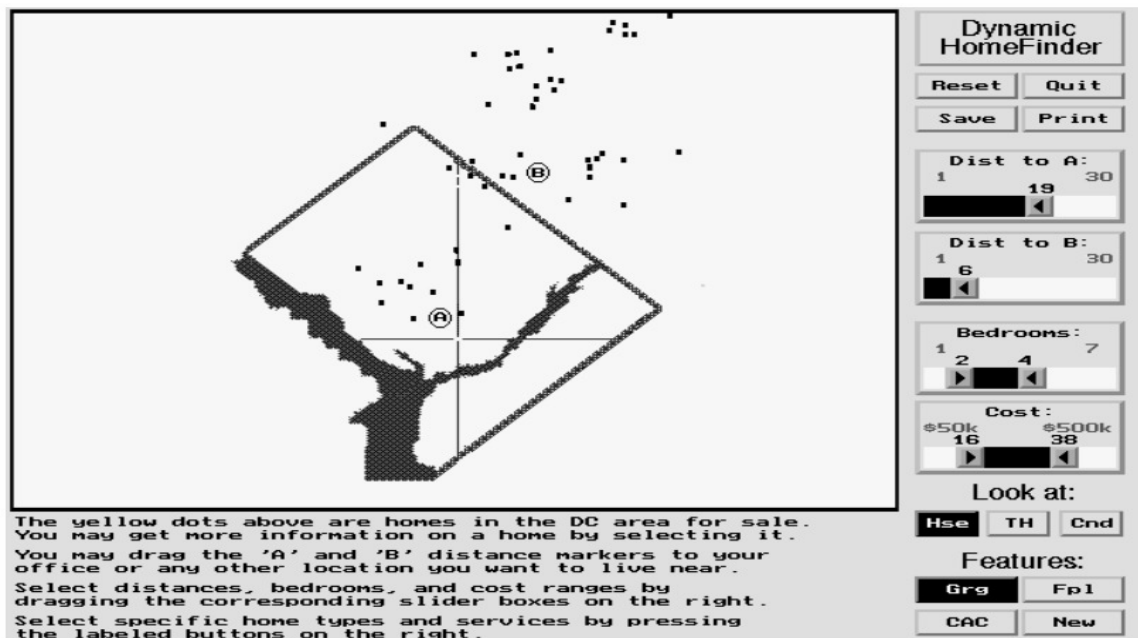
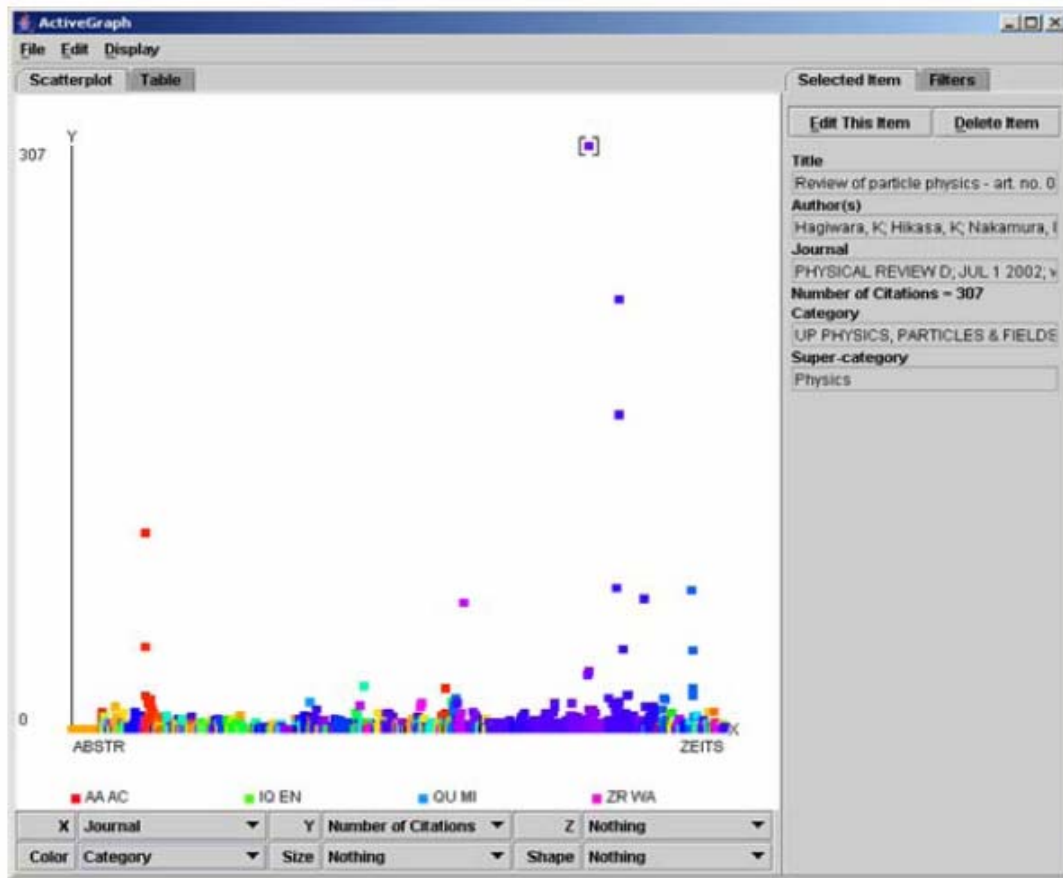


Figure 3-1: Dynamic Homefinder(Williamson 1992)

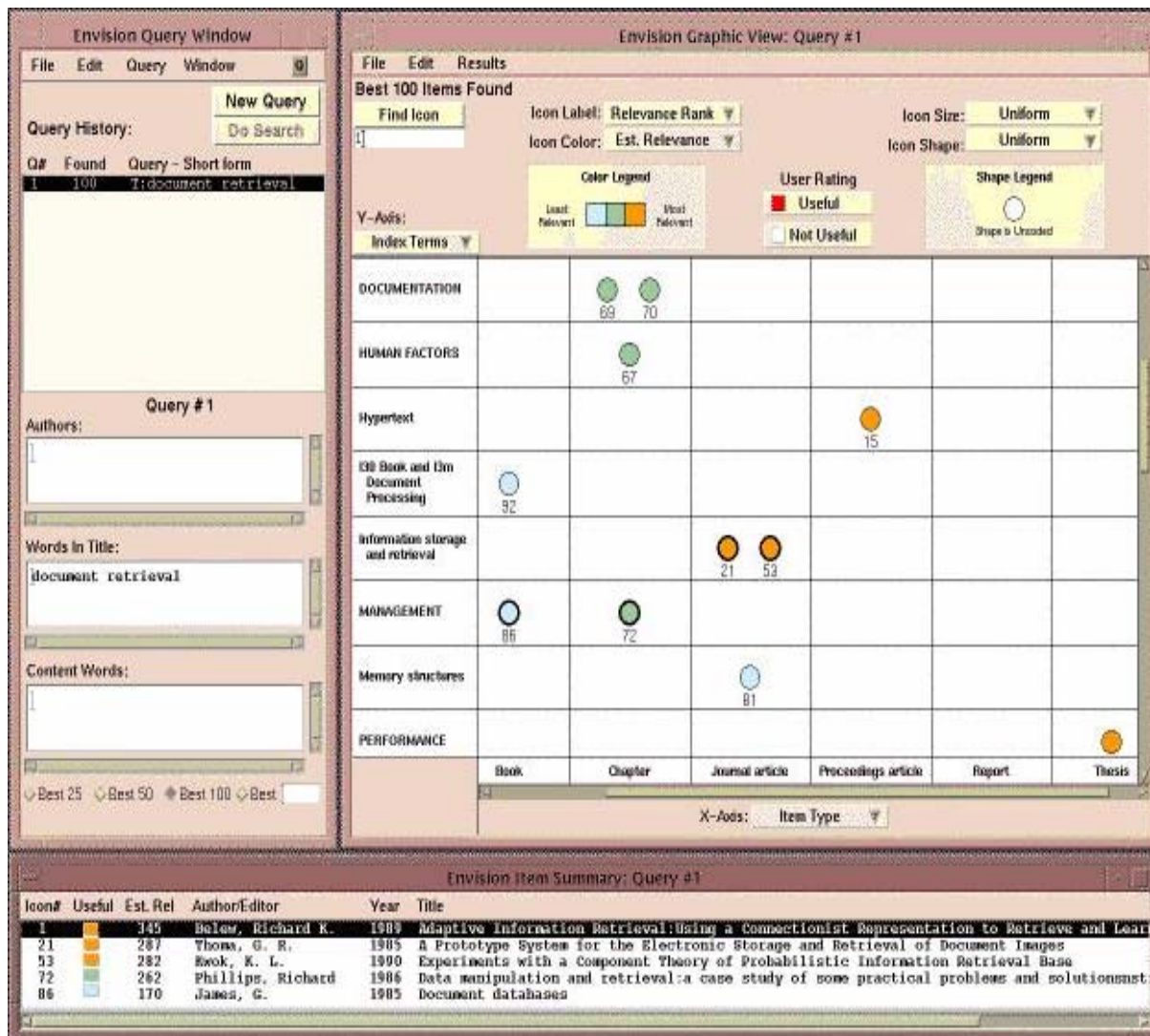
### 3.1.2 ActiveGraph and Envision: document collection visualization

Visualization tools in the Digital Library domain serve as visual search interfaces to databases of meta-data about documents. Envision (Lakkaraju 2004) and ActiveGraph (Marks 2005) are two examples of document collection visualizations. The figures 3-2

and 3-3 show these interfaces. In both, documents are the data objects. Some attributes of these documents may be “journal in which it was published”, “year of publication”, etc. A detail view shows meta-data information for any selected document. Again, **these two interfaces also deal with shorter attributes and lesser number of attributes than Datamaps.**



**Figure 3-2: ActiveGraph Visualization of search result sets(Marks 2005)**



**Figure 3-3: Envision for digital libraries (Lakkaraju 2004)**

### 3.1.3 Mail-Sleuth: email visualization

Mail-Sleuth (Eklund 2004) is an email visualization interface. Figure 3-4 shows a screen-shot of Mail-Sleuth. Usability studies show that with design adjustments, novices can read lattice diagrams without requiring any previous training in reading lattice diagrams, mathematics or computer science (Eklund 2004).

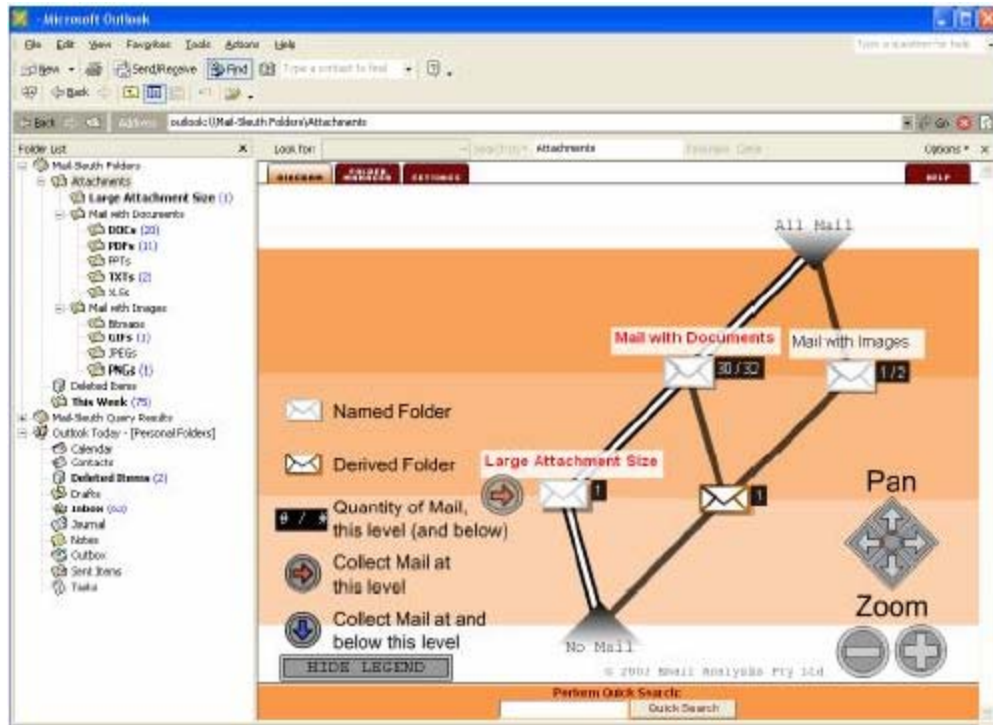


Figure 3-4: Mail-Sleuth for email visualization(Eklund 2004).

### 3.2 Discussion of visualization tools for novices

There is evidence that **novice users can work with advanced visualizations in the first session, provided there is good interface design.** The tools discussed involved dynamic querying to locate suitable houses, line diagrams to manage email and visualizations of document collections. However, **each tool is unique, and Datamaps has its own design challenges** in terms of size of data set and the difficulty of understanding long data attribute names.

### 3.3 Designing for novices: techniques and previous research

Specific techniques used to address novice user issues are described and previous research in this area is discussed here.

### 3.3.1 Integrated instruction technique

Neerinx and de Greef, in an experiment, show that help systems do not assist users in learning how to perform tasks (Neerinx 1993). They suggest that users learn how to use a system more efficiently and with better recall if they learn to perform tasks through integrated instruction. Neerinx et al. tested integrated instruction designs by adhering to two rules: 1. Integrate aid into the task to provide constant direct communication (as opposed to the indirect communication of the help system). 2. Model the aid on an expert model of the users' task to ensure that the flow of the aid is correct.

### 3.3.2 Exploration technique

John Reiman (Rieman 1996) conducted a field study with novices learning to use interfaces. Their learning behaviors were recorded over time in diaries. Exploratory learning occurred on a need-to-know basis (rather than for pleasure or learning for the sake of learning). It also occurred in conjunction with other forms of learning such as emailing someone for assistance, watching an expert use the interface, help documents and online help. However, **in this work we consider only a single novice user, so this discussion is limited to the pros and cons of exploration, help and tutorials** (and does not include asking others).

An immediate benefit of exploratory learning is that users only learn what they need to learn and are **not forced to learn unneeded topics like in demos** or sequenced tutorials. These weaknesses mainly deal with **users not exploring enough** to understand the complexities of a system.



### 3.3.3 Animation technique

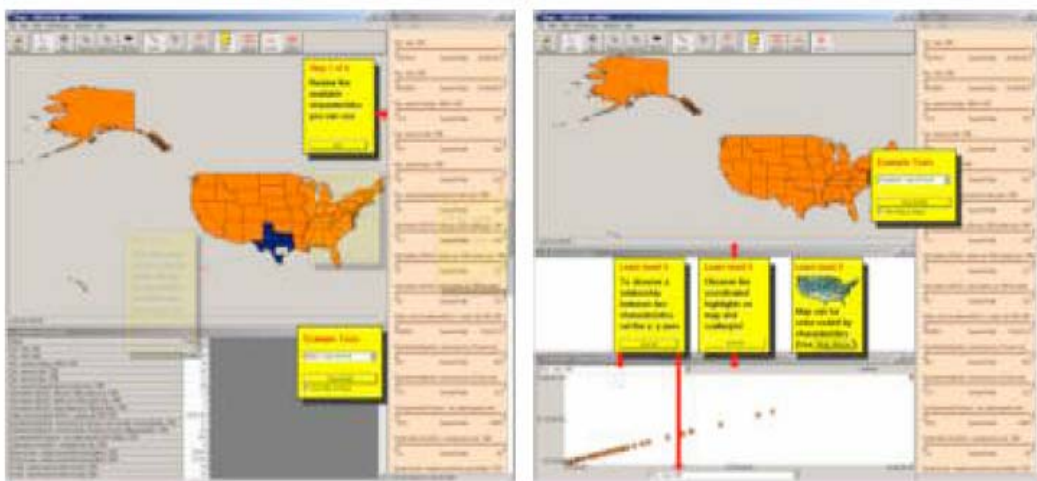
Animated step-by-step procedures are sometimes encountered as a technique to help novice users. Tversky (Tversky 2002) presents a survey of literature on using animation to educate physical phenomena, mechanical or biological systems, computational concept and interface usage. The important question is – what things to be taught (and learned) are being animated? In other words, is the animation merely superficial; could it have been the design that was successful? While animation might have been successful at conveying certain kinds of concepts, Tversky et al speak about failures in using animation to teach interfaces to novices.

Three works have been discussed - the first by Payne et al (Payne 1992) aims at instructing how to use the Macintosh; the second (Dyck 1995) on how to use MacDraw and the third (Harrison 1995) on how to use HyperCard. In all of these experiments users in animated graphics condition did not fare better than those in equivalent text or static graphics conditions. They fared better than only those with no instruction at all. In Harrison's work, after a week's time, the users in the animated graphics condition fared poorer than the group that received textual instruction. This has been attributed to deeper processing of textual instruction than animated ones. **An animation is fleeting; it might be difficult to perceive it, understand it and then use the lessons learned from the animation to the current context.**

### 3.3.4 Sticky notes

Sticky note help are special kind of help document. The instructions are broken down into little bits and are physically placed near the portion of the interface that it is addressing. Figure 3-5 taken from the work of Kang, Plaisant and Shneiderman (Kang

2003) shows sticky notes. Sticky notes were first implemented in the PhotoFinder kiosk (Shneiderman 2002). Integrated Initial Guidance (IIG) implements sticky notes inside the interface itself, thereby allowing users to use the interface or run automated demonstrations while reading the sticky notes overlaid on the interface. However, Kang et al. speak of the difficulties in designing these. Locating sticky notes without **obstructing crucial screen space, grouping which instructions go on which note, and help going out-of-synch with what users do – these are the serious problems**. There is a good chance that users may not exactly follow what is written, and get the interface to a state where they become unusable.



**Figure 2:** Two examples of IIG sticky notes. The left example (a) corresponds to a serial grouping of sticky notes. It shows the series of steps to accomplish a task. The right example (b) shows the sticky notes representing separate or “parallel” features in a level of the interface.

### **Figure 3-5: Sticky notes for integrated initial guidance**

The old Datamaps had sticky note help. However, these sticky notes would be invoked based on a chosen question. In one case, a user was trying to find the answer to “Of the western states, which has lowest unemployment rate?” He chose “How do I learn about one area?” under help questions, and it popped up sticky note help. It gave

procedural instructions about how to choose states on the map and view values on the table. However, he had wanted to know the answer to “which states qualify as western states”. He had expected the answer to be along the lines of which group of states are called “western”, which groups are called “mid-west” and so on. This example reveals that the **success of sticky notes depends heavily on the user landing the right question that invokes the right sticky notes for the situation.**

### **3.3.5 Complexity in levels or layers**

Shneiderman advocates the use of a “multi-layer” approach (Shneiderman 1998). Each level has predetermined features and can be designed so that **first time users or novices are exposed to layers in increasing complexity.** McGrenere, Baecker and Booth’s technique (McGrenere 2002) is similar but allows a variation to this by allowing users to specify which features go in which layer. The above example (Figure 3-5) is not only an illustration of the use of sticky notes, but also an example of the “multi-layer” approach. Both techniques were used in that interface.

## **3.4 Discussion on the techniques and previous research**

**Help, animations, sticky notes, sequenced tutorials are all extrinsic methods** of teaching a first-time user. Sticky notes appear on different windows or planes (spatially extrinsic). Help documents rarely explain the current situation in the interface – there is an additional step of adapting instructions to current situation (contextually extrinsic). Several authors show that users rarely consult help. Extrinsic methods of teaching novices may be helpful – but design that is easily understandable without these extrinsic aids will be superior.

**The focus of this research shall be on design features that can intrinsically increase understandability.** The design techniques such as Shneiderman's multilayer approach that rely on **exposing novice users to increasing complexity seem promising as intrinsic design techniques.** Also, there is evidence in novice behavior literature to suggest that novices are greatly influenced by display (Mayes 1988; Howes 1990; Howes 1991; Payne 1991; Kitajima 1992). Novice users explore based on trial and error. These explorations are driven by what they can see on the interface. Therefore, **there shall be emphasis on “making things visible”.** Whatever is worth exploring should be made prominently visible.

## 4 Learnable Visualization Elements Framework

### 4.1 Cognitive Artifacts

Norman defines a cognitive artifact as an artificial device designed to maintain, display, or operate upon information in order to serve a representational function (Norman 1991). It changes the way in which a user performs his task. It does not necessarily improve the unaided human's abilities to achieve the goal, but when present as an external aid in addition to the user's "Knowledge in the Head" it eases the achievement of the goal by changing the nature of the task/steps to be performed to achieve the goal. Relational data visualization tools may be called a cognitive artifact in the sense defined by Norman. For example, let us consider the following scenario:

*Mary has to give a talk about percentages of racial/ethnic groups that make up the whole population of California. She has made a presentation. If she uses a pie chart (and other visualizations) in the presentation, her task is simplified. She does not have to remember the percentages. All she needs to know is how to read a pie chart (**Knowledge in the Head or KIH**). The rest of the information is contained in the chart (**Knowledge in the World or KIW**). From Mary's point of view, the nature of the task she has to do to prepare for the task has changed, she has to prepare and "read" the chart, not memorize the ethnic group percentages.*



**Figure 4-1: Visualizations are cognitive artifacts**

## 4.2 Effective distributed cognition systems

Norman says that a person plus an artifact is more efficient because the Knowledge is distributed. The artifact contains “Knowledge in the World” (KIW), the person carries “Knowledge in the Head” (KIH). “Knowledge in the world,” reduces the amount of “Knowledge in the Head” we need to carry around. This is the “system view” or the view of an objective person considering the system of user and artifact. From the users’ point of view, the nature of their task is no longer monotonous or burdensome. For example, if a person is using a to-do list as an artifact, that means that the task has changed from remembering items to making and referring back to the list.

Knowledge in the artifact is usable for a task only if the user can interpret it. For example, if a person is trained in abacus arithmetic he can use an abacus for multiplication. The beads only complement the procedural and operational knowledge in the head. In short, Knowledge in the World must be a complement to Knowledge in the Head, for a distributed system. This balance between the two gets greater significance with more complex artifacts, as in the bar chart example in the table. Table 4-1 shows some examples “balance” in the distributed cognition system.

<b>Knowledge In the Head</b>	<b>Plus Knowledge In the World</b>	<b>Achieves the task:</b>
- Alphabetization - Reading skills	- Dictionary	- Looking up a word
- Training in abacus arithmetic	- Abacus	- Arithmetic calculation
- Basic arithmetic	- paper and pencil writing	- Arithmetic calculation
- Training in using a calculator	- Electronic calculator	- Arithmetic calculation
- How to read a bar chart - How to read a color legend	- Color coded bar chart	- Insights based on bar height (insights based on color meanings may be missed or not understood)
- How to read a bar chart - How to read a color legend	- Color coded bar chart - With legend explaining color code	- Insights based on bar heights - Insights based on bar colors

**Table 4-1: Knowledge in the World and Head must complement each other**

### 4.3 A user plus visualization tool system

Designers of everyday objects like analog clocks and dictionaries – have the luxury of assuming that people have been trained on how to use them. For complex artifacts like Visualization tools, users cannot be trained before hand and may not possess all the knowledge required to use all of its features before hand. In a visualization plus novice user system, it is important to consider:

- Knowledge In the World of Visualization– The designer’s goal is to make the artifact serve as knowledge in the world for several tasks. The visualization tool must communicate enormous amounts of data facts that exist “in the world”.
- Knowledge in the Head of Novices – What the target audience has been “trained in” or “used to doing”.
- Bridging the gap - *Design must communicate Knowledge in the World to the user, in terms of Knowledge in the Head.*

### 4.4 Need for a framework of learnable elements in a Visualization Tool

A visualization tool is an entire interface, providing a variety of visualization components and mechanisms to interact with them. Each of these is “Knowledge in the World”. Each of these is a learnable element within the cognitive artifact. Design must make these elements “learnable” in the very first usage session.

“All data values are non negative” and “Darkness of color means greatness of value” are two examples of learnable elements of very different kinds. Simply treating “all the Knowledge contained in a Visualization tool” as a single concept may not be useful during design. It would be more useful for designers to have a framework of types of learnable elements (pieces of knowledge). **Using this framework of “learnable visualization elements” helps to see how learning of specific aspects of the tool leads a novice user into adopting visualization-based strategies.**

## 4.5 Taxonomies for Information Visualization

Before forming a framework of knowledge contained in visualization tool, several taxonomical works are considered. Although these taxonomies aim at classifying information visualizations, **the criteria used for taxonomical classification reveal features of visualizations that potentially look new and must be learned by a novice. So these taxonomies may be used to build a framework of learnable elements contained in a visualization tool which must be learned.** Tweedie (Tweedie 1997) characterizes interactive externalizations (visualizations) along 5 dimensions:

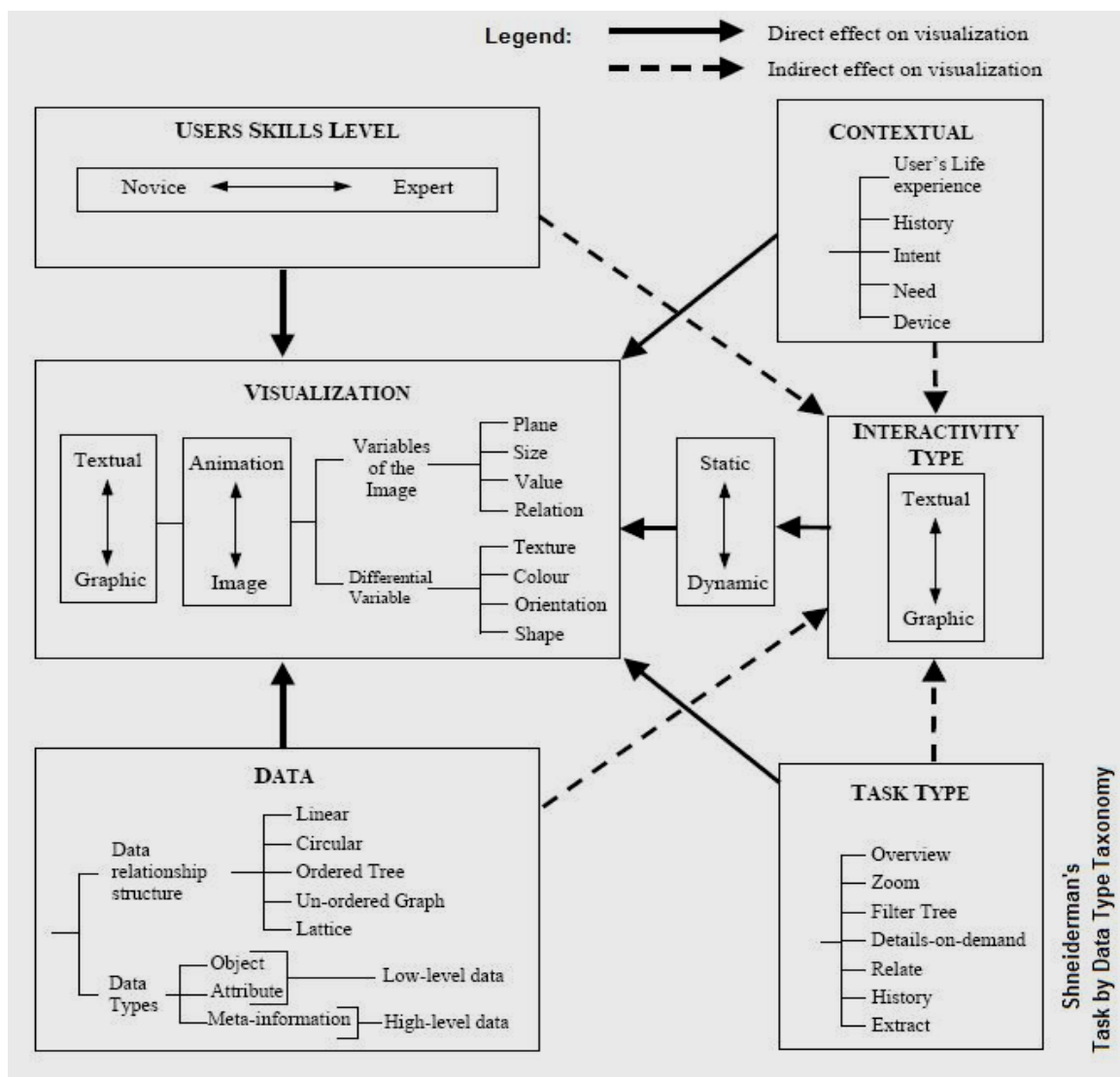
- The purpose or objective behind the visualization,
- what is visualized (data values or data structures),
- representation (the visual mapping or the graphical grammar used)
- interactivity (on a direct manipulation, indirect manipulation continuum)
- input-output representation as defined in Draper (Draper 1986)

Pfitzner, Hobbs and Powers (Pfitzner 2003) propose a taxonomical framework (illustrated in figure 4-2) to classify visualizations based on five interface design factors:

- Data – including data structure, data values and meta data



- Task – what the user is trying to achieve and the mechanism to achieve it. Within this item, the seven kinds of tasks according to Shneiderman’s “Task by data type taxonomy” are described (Shneiderman 1996).
- Skill level – does the visualization address the needs of novice/expert users
- Interactivity – interactive or not, and if interactive, where does the mechanism lie in the “direct manipulation - indirect manipulation” continuum.
- Context – external factors such as device on which visualization is displayed



**Figure 4-2: Pfizner’s taxonomy for Information Visualization (Pfizner 2003)**

Kennedy, Mitchell and Barclay (Kennedy 1996) provide a framework for describing an interface to a database system. The user, the database, the visualization and the interactions are part of the framework:

- The user component describes the human user of the system, providing a description of his sophistication (expertise with the data visualized or the system itself), the tasks which he wishes to perform, and his authority to view or modify data.
- The database component represents the actual database, encompassing the data model used, the schema of a particular data-store and the instances which populate it.
- The visualization component considers the presentation of the data to the user: the metaphor used to represent the items of data, and the physical layout on the screen.
- The interaction component addresses how the user may alter the presentation of the data, or the data itself. This comprises the user's intention, the medium selected to realize the intention, and the effect of the action. This effect may change the data, the visualization, or both.

North considers “data” “visualization” and “coordinations” as key elements in a coordinated-view visualization tool (North 2000). In his taxonomy of Coordinated-View Visualization tools, he says that they may be categorized by how much flexibility they offer with respect to:

- **Data:** can the user import their own data to be visualized
- **Visualization:** can the user choose what types of visualization
- **Coordinations:** can the user specify what kinds of coordinations they want between pairs of visualizations.

## 4.6 Using taxonomies to build a framework of learnable elements in a Visualization tool

Based on the above taxonomies, a framework for knowledge contained in a coordinated-view visualization tool may be formed. This shall be called the “Learnable Visualization Elements Framework”. The taxonomies examined are at either:

- **Interface level:** Both North’s and Kennedy’s taxonomies speak broadly at an “entire interface” level: the whole visualization tool with its constituent visualization components.
- **Or, Visualization component level:** Tweedie’s and Pfitzner’s taxonomies deal in detail with single visualizations.

The Learnable Visualization Elements Framework shall have the following features:

- **Addresses both interface-level and visualization-component-level elements:**  
This framework deals with listing out learnable elements in a coordinated-view multi-dimensional relational data visualization tools. Such a tool consists of several individual visualization components that may coordinate with each other. This is why both interface-level elements and individual visualization-component-level elements are included in the framework.

- **Focuses on learnable elements:** The framework shall answer the question “What are the things to be learnt in the visualization tool for a novice user to effectively use the knowledge contained in it?” Each element is one of the “things to be learnt”.
- **Borrows from previous taxonomies:** The framework shall be based on the above taxonomies, but modified for the purpose of listing out learnable elements rather than classifying information visualizations.

## 4.7 The Learnable Visualization Elements Framework

As in North’s Taxonomy, the top level elements in the “Learnable Visualization Elements Framework” include Data, Visualizations and Coordinations. “Overall interface level controls” is added as a fourth item for completeness:

- Data
- Visualizations
- Coordinations
- Overall Interface level controls

User-related elements (as in Pfitzner’s or Kennedy’s taxonomies) are not included, since the focus is on the “Artifact” portion of the distributed cognition system.

### 4.7.1 Data

First, we shall consider “Data”. As in Pfitzner’s taxonomy, “Data attributes”, “Data objects” or “Tuples”, “Data relationship structure”, and “Meta-data” are defined. One more item, “Data values” is added. This may help distinguish cases where users may be

familiar with attributes alone, but not with the values. For example, consider a database with a data attribute “Inhabitant Fauna” about Data objects “countries of the world” – although users may know what “inhabitant fauna” means, they may not know about particular values (species of animals). So the list of elements under “Data” is as follows:

- Data
  - Data attributes
  - Data objects
  - Data values
  - Data relationship structure
  - Meta-data

#### **4.7.2 Visualizations**

Visualizations have a grammar or visual mapping rule. Data objects are represented as visual marks and the properties of visual marks show the data values. The mapping or visual syntax must be understood to obtain insight about the data.

Merely knowing the syntax is not sufficient. There is a higher level knowledge required about the visualization. What is the visualization showing? How do all the visual marks relate to each other? A semantic knowledge is required to identify meaningful patterns and trends.

We consider interactive visualizations – that is those that allow the user to perform operations on them. Tweedie and Pfizner have described interaction in terms of a “direct manipulation to indirect manipulation” continuum. Kennedy provides the distinction between the “user’s intention” and the “medium selected to realize the intention”. That

separates the conceptual notion of an operation from the mechanism that helps execute the operation. This separation is useful because it helps capture the difference between knowledge about what *can* be done and knowledge about *how* to actually do it. Without the distinction, it would be harder to point source of learnability/understandability problems. Consider the following scenarios:

*“I am almost sure this visualization should allow me to select a unit, I am just not sure how to do it. Perhaps I should try the buttons?”*

*<This is some one who knows what to do but has just not yet figured out how to do it>*

*“I’ve clicked this button and accidentally discovered that you can select a unit. I didn’t know that before...”*

*<This is some one who hadn’t known that selection could be done at all. Before this discovery, the first person was in a better position than this person. >*

The following elements are defined under “visualization”:

- Visualization
  - Visual syntax
  - Visual semantics
  - Operations: Conceptually allowed actions on visualizations. This is equivalent to Pfitzner’s definition of “Tasks” on a visualization. However, “operations” has a lower level definition. So it includes more actions than in the task taxonomy. Even a low level action such as choosing different color palettes is an operations on a visualization tool.
  - Interactive mechanisms or Operation Execution mechanisms: widgets or mechanisms that allow execution of the operations. “Operations” and “interactive mechanisms” are distinguished rather than treating them both as one unit called “Interaction”

### **4.7.3 Coordinations**

A visualization tool is composed of several coordinated visualization components.

A coordination is an action, invocation pair (North 2000). An action on one of the visualizations may invoke an action in another visualization. The coordinations between views must be learned or understood in order to use them for tasks.

### **4.7.4 Overall interface level controls**

Print commands and other general menu items, window positioning or resizing commands, labels titling different components in the interface, saving options, user settings and preferences etc. are some examples.

## 4.8 Learnable Visualization Elements Framework Definitions

Table 4-2 summarizes the Learnable Visualization Elements Framework.

<b>Data related</b>	<b>Data attributes</b>	Elements of description about data objects (or similar definitions in Relational Database terms)
	<b>Data Objects</b>	An entity with attributes (or similar definitions in Relational Database terms)
	<b>Data values</b>	Values that are associated with the different numerical or categorical attributes relevant to a problem. (as defined in Relational Database terms)
	<b>Data structure</b>	Relational structures in the data that characterize the data as a whole.
	<b>Meta-data</b>	Information about the data
<b>Visualization component related</b>	<b>Syntax</b>	Rules that define how a data value is represented as visual element
	<b>Semantics</b>	The meaning, purpose or objective of the visualization.
	<b>Operation Rules</b>	Conceptually defined actions that allowed on individual visualizations.
	<b>Operation executing mechanisms</b>	The widgets or mechanisms used to execute the conceptually defined operations.
<b>Coordination</b>		Rules that define action-invocation pairs: an action on one visualization invokes an action in another coordinated visualization.
<b>Interface level Controls</b>		Menus controls and commands provided at the interface level.

**Table 4-2: A framework of learnable elements in a visualization tool**



## 4.9 Knowledge in the Head for target users

Now we move on to the “Knowledge in the Head” part of the distributed cognition system. In this work, we consider beginners learning and understanding a relational-data coordinated-view visualization tool in the very first usage session. The user is assumed to have only:

- GUI experience: Experience with graphical user interfaces
- General knowledge: ability to communicate in natural language, ideas about political boundaries on maps etc.
- Familiarity with graphical representations
- No expertise in data set or information visualization is assumed.

## 4.10 Difference between a novice and expert

“What is the Knowledge-in-the-Head of the user?” is an important question for a distributed cognitive system because users bring their previous knowledge to the new system, and make use of it. Rosson and Carroll describe how users carry their previous knowledge to work with new interfaces (Carroll 1987). Those aspects that distinguish an expert from a novice may be important points to note and to design for. The following are possible differences between novices and experts:

- **Knowing to look for data:** People who have collected, experimented with or compiled the data would be experts in the data set. They would be familiar with the data objects and attributes and data structure. However novices don’t have this expertise. **This is related to the “Data” element in the Learnable Elements Framework.**

- **Knowing that visualizations map several attributes onto visual properties:**

Experts who are used to data exploration may be expected to watch out for which visual cues mean what. A novice user may not even be aware of the fact that visualization components encode so much. **This is related to the “visualization” element in the Learnable elements framework.** The following scenario illustrates this problem (Susan’s political map atlas scenario):

*Susan, an elementary school geography teacher, uses several Atlases. In these Atlases, she has often seen states shaded with different bright colors for the purpose of easily distinguishing them even though the state borders serve the same purpose. She is using Datamaps for the first time. She is seeing a map colored by “population 1997”. She is not an “info-vis” expert. She can perceive colors of the states on the map. But to her, they are a means of distinguishing the states. Although she has seen choropleth maps on Atlases before, it does not occur to her in this case that color indicates value of the state. The Datamaps designers are trying to provide valuable information about population of a state through color.*

- **Knowing to choose coordinations by task:** A coordinated view visualization tool offers multiple visualizations. Pairs of coordinating visualizations may offer answers to specific questions. For example, the question “are all states with both high-income and high-education level states on the northeast” can be answered by using a coordinating map and scatter-plot. An expert is inquisitive, is used to asking insightful questions and forming knowledgeable plans for data exploration with ease. So for an expert, we can expect that choosing pairs of coordinations to

answer relevant questions would be fairly easy. As the Census usability study shows, these kinds of activities were difficult for novice users. **This is related to the “Coordinations” element of the framework.**

#### **4.11 Summary**

A visualization tool and user were considered as a distributed cognition system. Design must communicate Knowledge in the World (in the visualization tool) in terms of Knowledge in the Head (what a first time novice user knows.) For this purpose, a framework of types of Knowledge in the World was formed, known as the “learnable visualization elements framework”. The framework helps build a systematic checklist of things that must be learned in order to effectively use visualizations, for any particular visualization tool. Next, the knowledge in the head of novice users was considered. In particular, three major differences were considered between first-time novices and experts (those who have used interactive visualization tools before). The three differences considered are related to three major elements in the Learnable Visualization Elements Framework: Data, Visualizations and Coordinations.

## 5 Design Hypotheses

The following hypotheses are made, based on the discussion of the three key aspects of visualizations in which experts have knowledge and novices don't. The three aspects are "Data", "Visualizations" and "Coordinations." The three hypotheses are based on these aspects:

### 5.1 The "Data-first" approach and Hypothesis 1

In several cases, information visualization tools display visualizations packed together in an attempt to show as much of the data as possible. The aim is to provide a visual overview of as much as possible. The data first approach implies making the data prominent in the Gulf of Evaluation stage. Benyon states, "Data is probably the only thing that people have in common with computers. The "Data-first" approach takes data, the common ground between people and computers, and makes this prominent in the Gulf of Evaluation stage. The data-first approach is:

- Provide an overview of all available data at start-up
- Provide clear distinction between the views that show structural relationships among data attributes (the data-attribute space) and the views that show visualizations of data values (the data-value space)
- Provide a mechanism to choose data attributes from this view so that they may be visualized
- Provide immediate feedback about which attribute has been chosen
- Provide a mechanism to input data attributes into visualization components with minimal user actions

**The first hypothesis is that the Data-first approach will lead novice users to adopt visualization-based strategies.** This is because data are the most familiar elements, and displaying them prominently will help match chunks of information needs to chunks of information visible on the screen as data attributes. For example, a person wanting to know about the “state with maximum value for unemployment rate” might find the chunk “unemployment rate” among the data when data attributes are prominent displayed.

## **5.2 The “Less Visualization for More explanation” approach and Hypothesis 2**

There exists a tradeoff between being able to pack in more visualizations and devoting more space to explain what the visualizations mean. The more the space devoted to packing in visualizations, the lesser the available space to explain what the visual marks mean, and the lesser the ease of understanding them in a first-time usage session. The “Less Visualization for More explanation” approach is to:

- Make space to provide conspicuous verbal explanations of visual syntax or semantics if visual cues themselves aren’t self explanatory
- Provide contextual explanations that are based on current interface state.

**The second hypothesis is that the “less visualization for more explanation” approach will lead novices to adopt visualization-based strategies. This is because the explanations help in understanding visual syntax.** Contextual explanations will adjust the explanation after obtaining interface state information. These explanations may be legends, labels, sentences which describe user actions and results.

### **5.3 The “Predetermined Task Based Coordinations” approach and Hypothesis 3**

Usually, the objective in relational data visualizations is to have high scalability.

Designers try to display as much as possible or as many coordinated visualizations as possible. Although this gives a good visual overview for experts who are used to seeing information visualizations, this design is a source of great screen complexity for novices.

The “predetermined task-based coordinations” approach is to:

- Provide predetermined set of visualizations on separate screens
- Provide coordinated visualizations according to task. Different coordinations support different kinds of tasks
- Provide a mechanism to navigate between or invoke the different predetermined configurations
- Provide the user with increasing complexity – expose them to less complex coordinations by default and let them go to higher complexities later on. (It is possible to provide this since there are predetermined configurations; the designer can choose to make the simplest coordination as the first thing a novice sees)

**The hypothesis is that the “pre-determined task-based coordinations” approach will lead novices to adopt visualization-based strategies in the first usage session.**

This is because users are given predetermined configurations labeled with the tasks that they help perform. Since they are exposed to less complex screens in the beginning, they can learn in increments about how different coordinations help achieve different tasks.

## 6 Design of Datamaps

First, the design of new Datamaps is described. This is followed by a description of the major iterations that led to the new design.

### 6.1 Description of final design

Figure 6-1 shows Datamaps at start-up. The list of all available data attributes are made prominently visible (Data-first approach). There are different icons at the top of the interface, showing the different predetermined coordinations (predetermined task-based coordinations approach). The predetermined coordinations are called “Available configurations” rather than “Available Coordinations” in the interface to better resemble everyday language. The simplest coordination is visible at start up, by default.

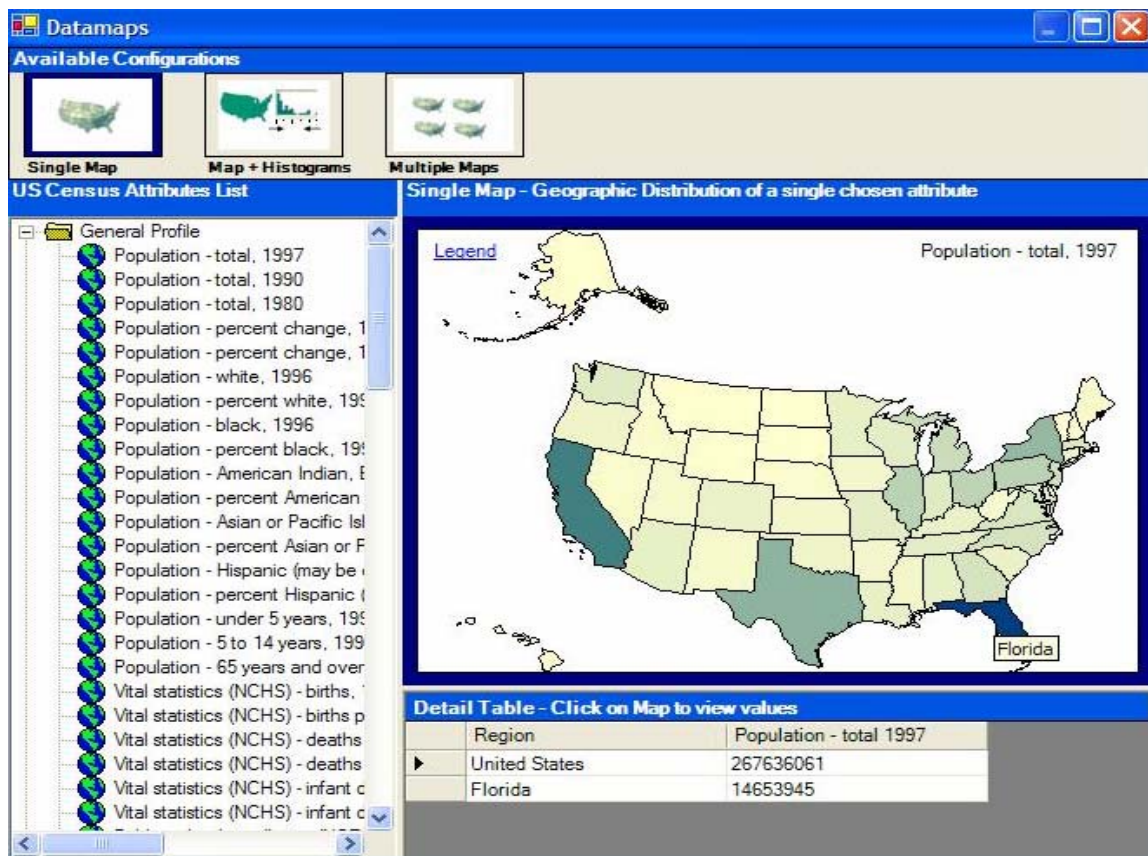
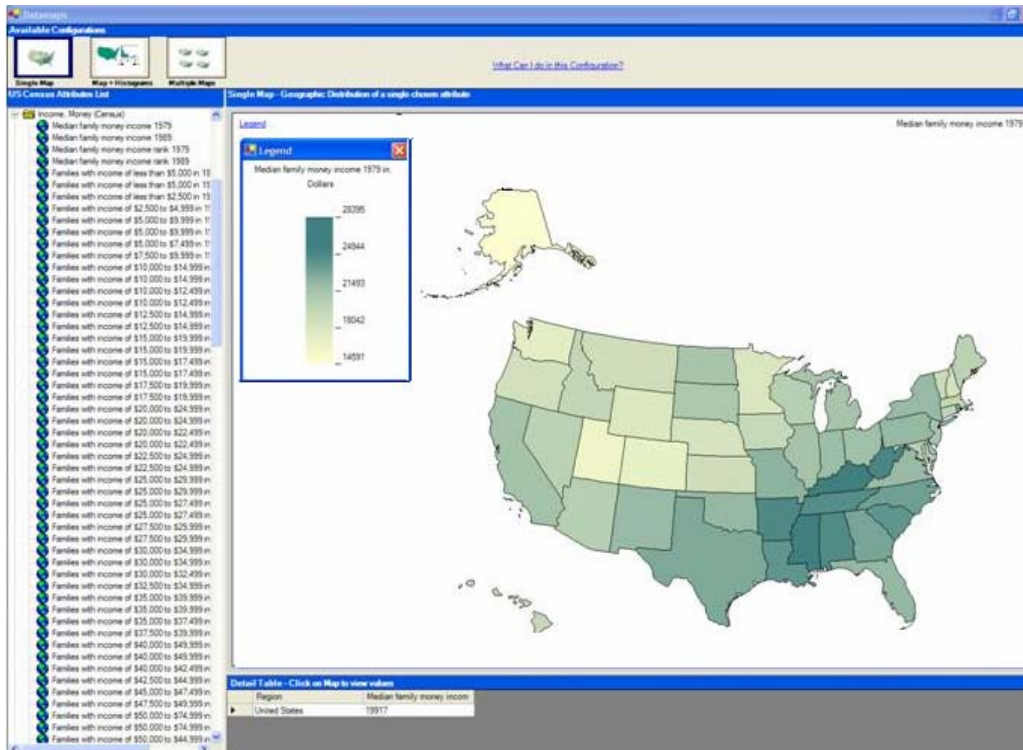


Figure 6-1: New Datamaps at start-up (Single map configuration)

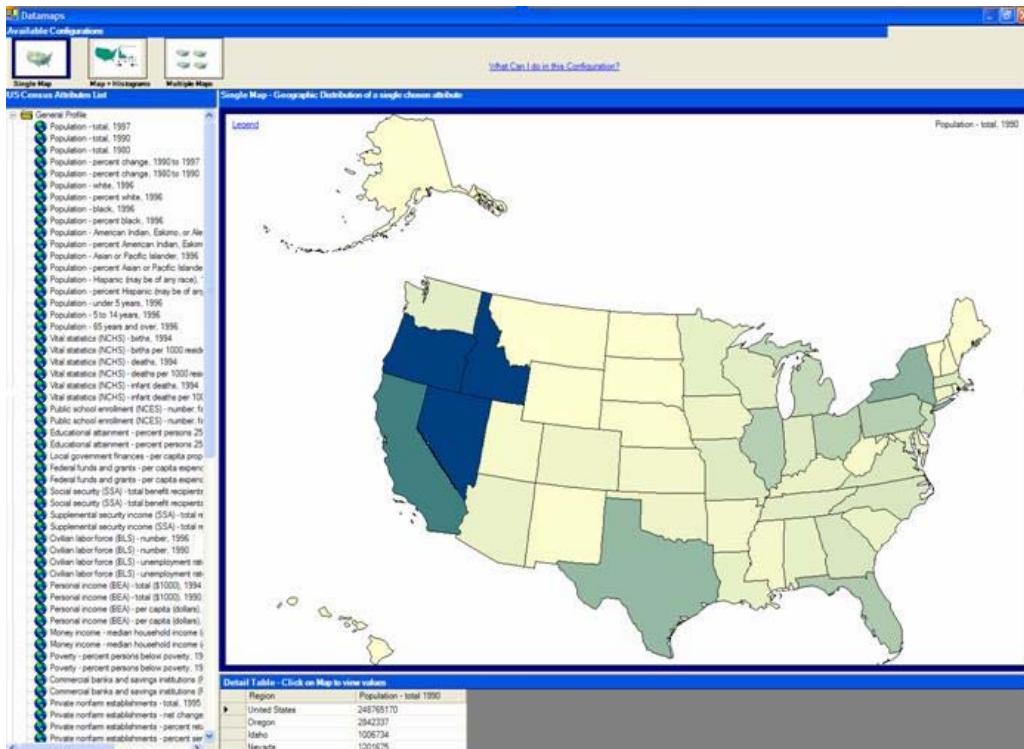
There is a link to the legend for the map. Upon clicking this link, an explanation of the colors shows up, along with meta-data. Figure 6-2 shows legend window for a map.



**Figure 6-2: Map with legend window open**

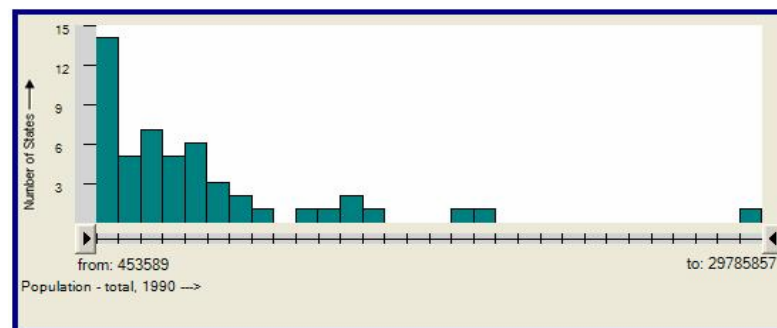
In the “single map” configuration, a single map is coordinated with the table. This configuration is for the task of viewing geographical distribution of a single attribute. Every time the user clicks on one of the attributes from the tree-view on the left, a new map is loaded. The map is a choropleth map, which appears colored by default. It does not require any further configuration to show colors. Clicking on a state on the map causes the table to load the value for that state into the table. Selecting several states on the maps, loads all the values into the table. Figure 6-3 shows the table loaded with values for several states. Unlike the original version of Datamaps, the table shows values only for the current attribute.





**Figure 6-3: Table shows values for all selected states for current attribute**

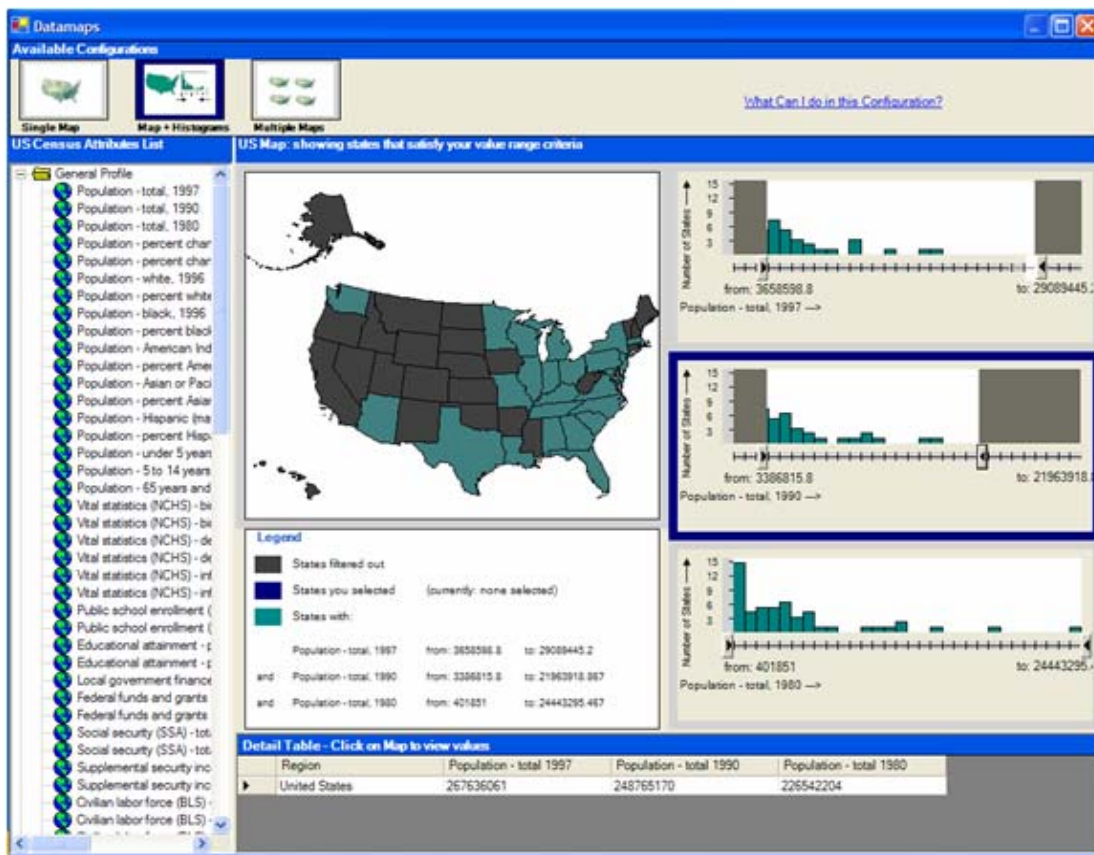
Upon clicking the “Map+Histograms” icon, another predetermined coordination shows up. In this configuration, a map coordinates with histograms through dynamic query sliders. The height of the bars in the histograms, stand for the number of states in a given class interval; this is marked on the y-axis. Figure 6-4 shows a histogram in detail.



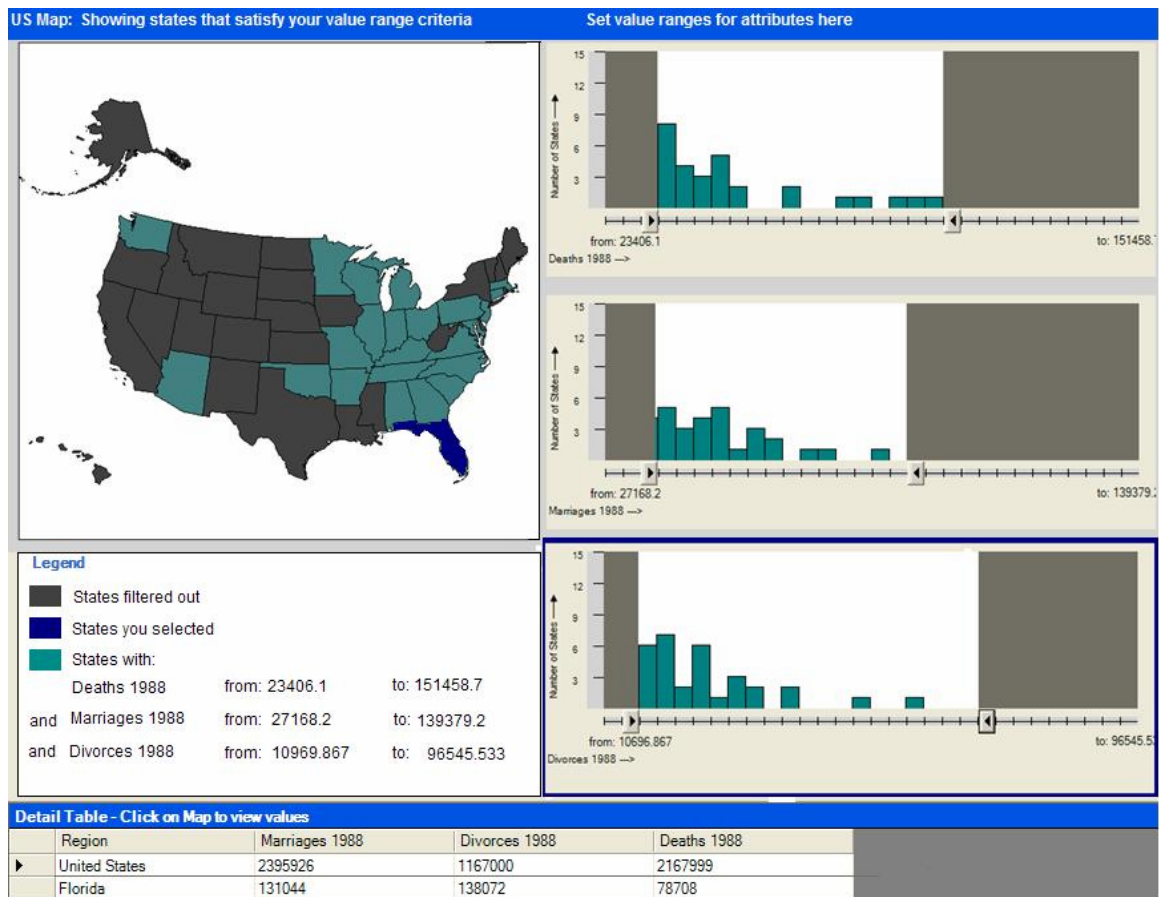
**Figure 6-4: Bar histograms**

The histograms are much larger and are more readable than in the original version of Datamaps. This was possible, partly due to the space gained by splitting up the available visualization components into separate configuration screens.

To specify the desired range, the user has to drag the dynamic query slider widgets for the upper and lower bounds. Figure 6-5 shows the “Map + histograms” configuration. The dynamic query slider widgets have been moved from their default positions for two of the histograms. This has deselected some of the bars. Correspondingly, states on the map have also been deselected in the same color. As query sliders are used to set range criteria for additional attributes, additional states are filtered out on the map. Only states that satisfy all the criteria remain green on the map.



**Figure 6-5: New Datamaps with “Map + Histograms” configuration**



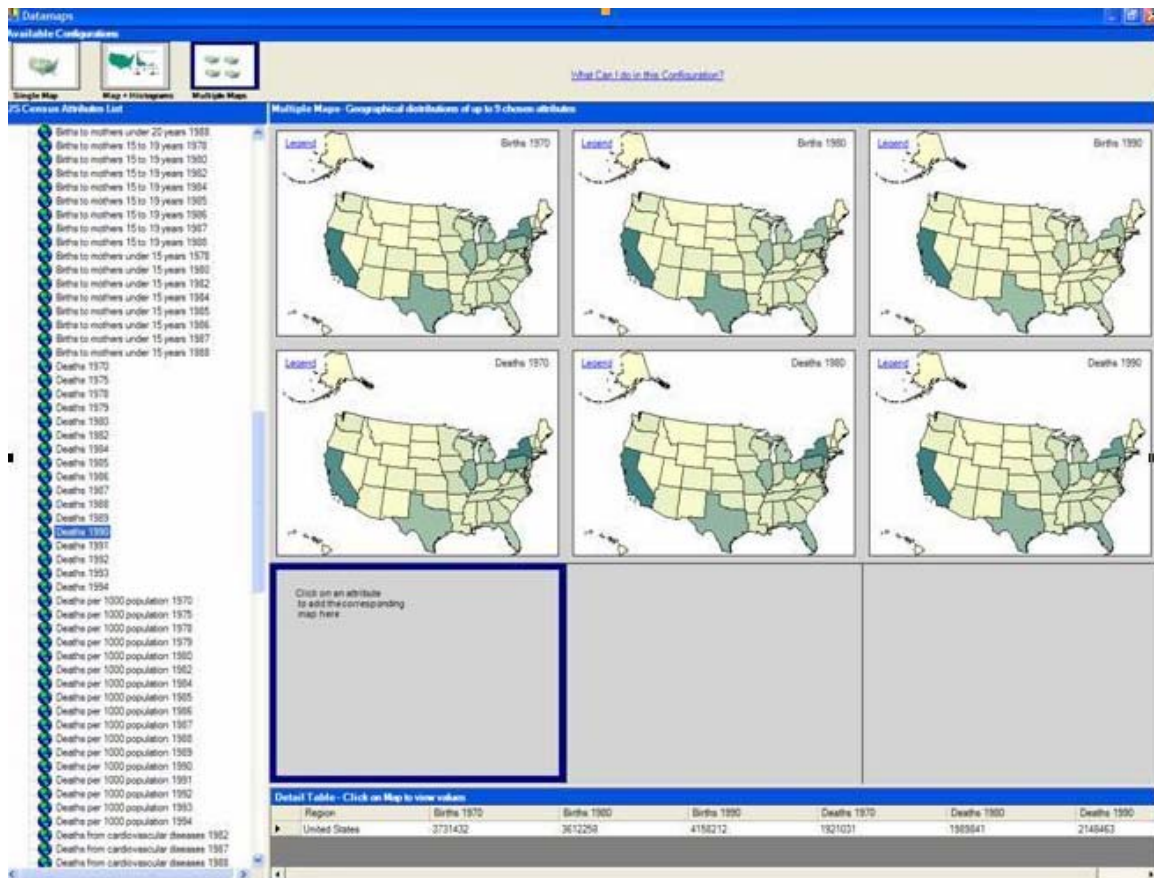
**Figure 6-6: The dynamic legend explaining visual syntax**

A dynamic legend shows what is happening in the interface in the “Map + Histograms” configuration. Figure 6-6 shows the dynamic legend below the map. Table 6-1 clarifies the operation of the dynamic legend with an example:

User Action	Response from interface
“Deaths 1988” is selected from the attributes tree-view	<p>“Deaths1988” histogram appears</p> <p>Legend now reads “States with Deaths 1988 from 2064 to 222599” against green</p>
A second attribute “Marriages 1988” is selected from the tree-view	<p>A histogram for “Marriages 1998” appears</p> <p>Legend now reads “States with Deaths 1988 from 2064 to 22599 AND Marriages 1988 from 4726 to 236628” against green.</p>

**Table 6-1: Example working of dynamic legend**

The “map+histograms” configuration has the dynamic query slider widget, which is a Coordination execution mechanism. As it is moved back and forth, states are deselected dynamically. The lower and upper bound figures in the legend also change along with the slider movement.



**Figure 6-7: New Datamaps showing “Multiple maps” configuration**

The multiple maps configuration is the third available predetermined coordination (illustrated in figure 6-7). Whenever an attribute is chosen, a new map gets added to the grid. Grid slots get filled up to nine slots. Up to nine maps may be viewed at a time. Viewing multiple maps help visually scan geographical distributions of several attributes of interest. When a state on any one of the maps is clicked, the table displays values for all attributes that are currently being displayed in the configuration.

## 6.2 Design iterations and rationale

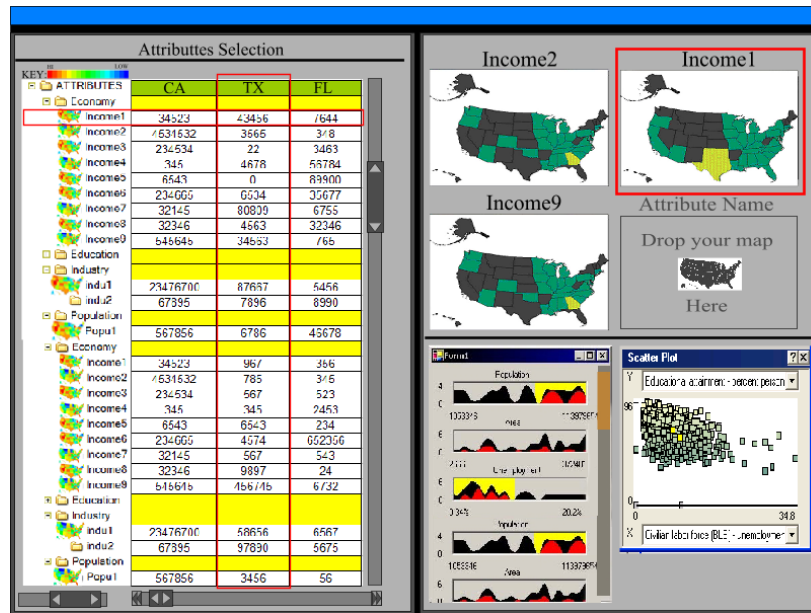
To arrive at the final design of Datamaps, several iterations and studies were conducted. Table 6-2 summarizes information from the three iterations.

Iteration	Hypotheses implemented	Description
1	<ul style="list-style-type: none"><li>• Data-first approach with drag-and-drop interaction design</li></ul>	<ul style="list-style-type: none"><li>• Data items are visible on tree view</li><li>• Attributes must be dragged and dropped on a visualization type, to view visualization</li></ul>
2	<ul style="list-style-type: none"><li>• Data-first approach with right-click interaction design</li></ul>	<ul style="list-style-type: none"><li>• Data items are visible on tree view</li><li>• Any attribute to be visualized must be right clicked. The desired visualization type must be chosen from the right click menu</li></ul>
3	<ul style="list-style-type: none"><li>• Data first approach with single left click interaction design</li><li>• Less visualization, more explanation tradeoff</li><li>• Predetermined task based coordinations</li></ul> <p>(All three hypotheses were implemented)</p>	<ul style="list-style-type: none"><li>• Data items are visible on tree view</li><li>• Any attribute to be visualized must just be clicked. Since the visualization is already in a predetermined “mode”, a new visualization is automatically loaded. The step of choosing a visualization is eliminated.</li><li>• The loaded visualizations are explained by static legends or dynamically changing legends (in the case of dynamic querying)</li></ul>

**Table 6-2: Summary of design iterations**

### 6.2.1 First Design Iteration

A new design and study were made by Yilmaz. A re-examination of the previous work by Yilmaz is described as the first design iteration of this work. Figure 6-8 shows a snap-shot of a mock-up of Datamaps designed by Yilmaz.



**Figure 6-8: An initial design using the Data-first approach**

This is what is called the “Data-first approach” according to this thesis. The objective is that users must be given an overview of all the data. The system shows all available data attributes, at start up, organized by categories. This design included the following features:

- Tree view of attributes by categories:** The Census Department has organized data attributes into a two level hierarchy: categories and attributes. Related attributes are organized into categories. This is visualized in the tree view. Figure 6-8 shows a tree of attributes organized by categories (leftmost)
- Table of values by state:** The tree view comes tightly coupled with a table of values by state as seen in the figure. Figure 6-8 shows values for California, Texas and Florida. Collapsing a category on the tree-view also collapses the table.
- Disabled visualizations at start-up:** Visualizations initially have a gray iconic display of and appear disabled. Figure 6-8 shows four maps; the bottom right map



is gray and disabled. When an attribute is input into it, an actual visualization is displayed and interaction is enabled.

- **Drag and drop interaction:** The desired attributes are dragged and dropped from the tree view into the disabled visualizations.
- **Selection highlights:** These reveal co-ordinations between the views. Figure 6-8 shows Texas selected on the “income1” map. Correspondingly, there is a red rectangle around the Texas column and another rectangle around the “income1” row. Table 6-3 lists pros and cons for each of these features.

Design Feature	Possible pros and cons
Tree view on the left	<ul style="list-style-type: none"> <li>+ Gives overview of all data available interface by categories</li> <li>+ Leads user to follow an action path from left to right, that is “Data attributes input to tree view section”</li> <li>- but shows attributes only by the Census Department’s Categorization</li> <li>- but involves having to open up “possible categories” to locate a desired attribute.</li> </ul>
Table of values by state aligns with tree-view	<ul style="list-style-type: none"> <li>+ shows values of selected states against open categories</li> <li>- but all cells of the table that are against closed categories are blank; this is waste of space</li> <li>- but user may not want to view values of other attributes in the open category; this causes unnecessary clutter</li> <li>- but user may want to compare values of two attributes which are very distant from the tree view</li> <li>- but may result in attracting too much attention to tabulations instead of visualizations</li> </ul>
Disabled visualizations at start-up	<ul style="list-style-type: none"> <li>+ Lead users to look at the more “alive” tree view section first</li> <li>+are similar to photo-editing software which have main work-space empty and the inputs or tools arranged around the main-workspace.</li> <li>- but screen real estate is taken up by inactive entities</li> </ul>
Drag and drop interaction	<ul style="list-style-type: none"> <li>+ Reinforces the idea that attributes are inputs to visualizations. In other words, reinforces the idea that “attributes are entities that can be visualized”</li> <li>+ is a highly direct manipulation technique</li> <li>- but tree-view elements in general do not have drag and drop interaction and may not be directly obvious to user.</li> <li>- but requires considerable use of the mouse to simply associate an attribute with a visualization</li> </ul>
Selections highlighted	<ul style="list-style-type: none"> <li>+ Makes it evident that there are relationships among the multiple views</li> <li>+ helps detect coordinated trends</li> <li>- but it may be an overload of information</li> </ul> <p>Since there are many views present</p>

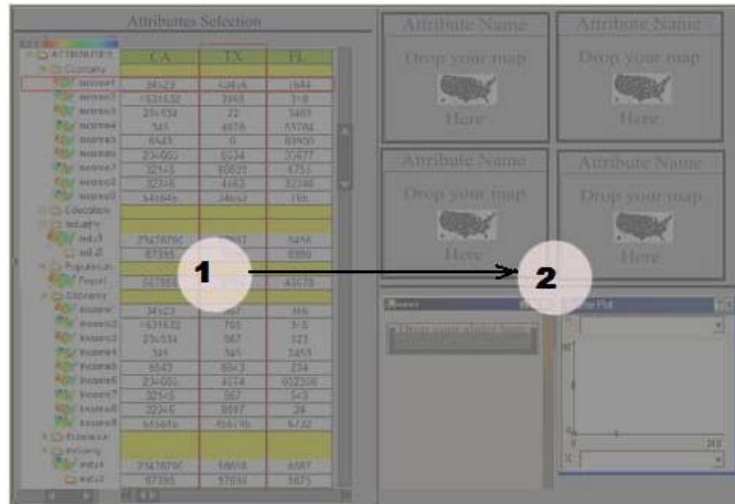
**Table 6-3: Claims for design iteration-1**

A study was done by Yilmaz on this design. Snap-shots of mocked-up screens were used for the experiment. The design was tested with a think aloud protocol. Four people in the 21-29 age range who fit the criteria for “novice user” were recruited as subjects. The test was conducted with a Wizard of Oz technique. Depending on user-inputs, the experimenter showed new snapshots of “would-be” interface states.

Users were generally able to organize their thoughts and actions from left to right, that is from “Data attributes to visualizations”. The table was easy to understand. However this made users solve several tasks using only the tables (This situation is what we seek to avoid). The coordinations between map and table were easy to understand. However, a critical incident that must occur before this coordination is understood - learning that attributes must be dragged and dropped on the visualizations. Users required hints about the drag-and-drop interaction, and this is not acceptable since this is the stage that affects the next stage of understanding – the coordinations. The histograms and coordinations between histograms were extremely difficult to understand. Finding out that dragging and dropping was difficult for all of them, and they required hints.

The most successful aspect in this design was the Data-first approach. Users clearly understood the structure of data – that is was grouped in related categories. In this design all five users exhibited a clear left to right flow in their action paths, illustrated in figure 6-9.





**Figure 6-9: New layout led users to organize actions-paths from data to visualization**

They concentrated on the left at the beginning of their interactions and then proceeded to the visualizations section. It may be recalled that users of the original Datamaps design had unclear about where to start their action paths. Without a clear visualization of the data-structure, they were lost in the interface. Here, it was clear that data was the starting point. Data are the most familiar aspects to novices, and a prominent display of data will most likely lead to the use of data as starting points of action paths.

However, the mechanism by which data is visualized (dragging onto disabled visualization components) was difficult to understand. The critical incidence here is the understanding of the label “Drop your map here”. That is what is directing the user to choose an attribute from the tree-view display of all available attributes, drag it onto the visualization section, and drop it onto the type of visualization desired. This was difficult to understand.

## 6.2.2 Second Design Iteration

In order to get rid of the problem with dragging and dropping, another design was proposed. The features of the new design were as follows:

- **Tree-View retained:** The tree-view was a success and hence it was retained
- **Right-click on attribute instead of drag-and-drop to view visualization:** An attribute on the tree view should be right clicked, for it to be visualized. When right clicked, a right click menu appears, showing all visualization types that can be used to visualize this attribute. If a visualization like scatter-plots is chosen, that requires more attributes in order to display a visualization, the system prompts user to select one more attribute before loading the scatter-plot. Figure 6-10 shows a right click menu; the user has right clicked on the first attribute. A large instruction in the empty area directs them to right click.

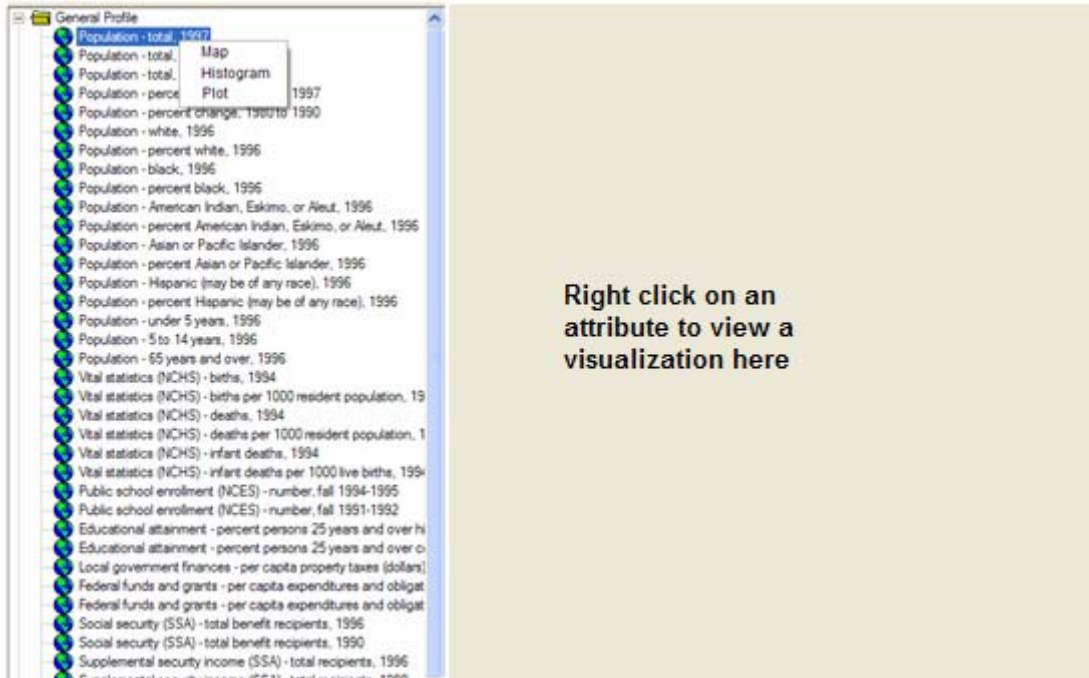
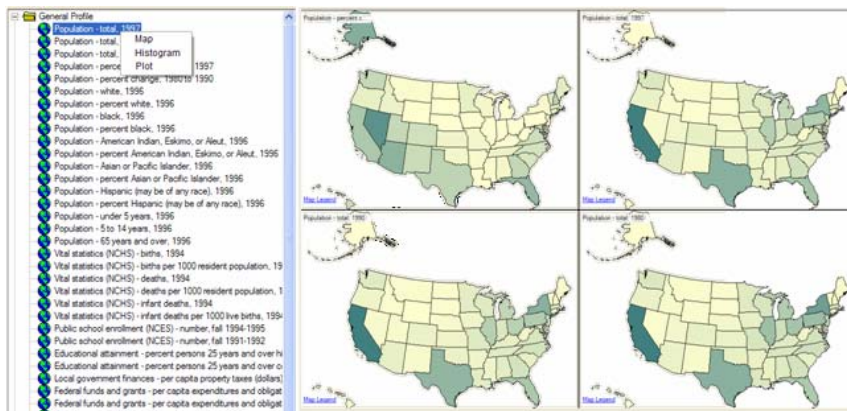
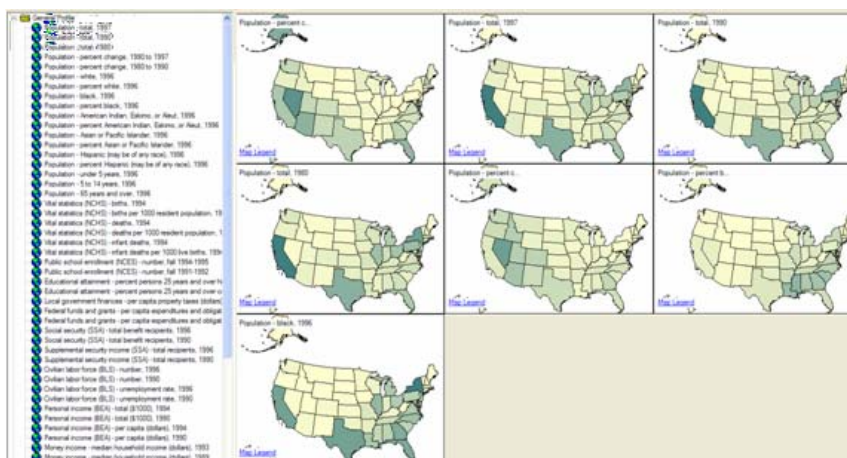


Figure 6-10: Right-click menu interaction to create visualizations for attributes

- **All visualizations present together, not grouped by visualization type:** Until screen capacity (nine visualizations) is reached, whenever a visualization type is chosen for any attribute, it is loaded on the screen. That is, a histogram for population 1997, a map for education 1990 may be on the same screen. The previous design had the visualizations grouped together by type and had predetermined locations – all histograms together, all maps together. This designed aimed at giving control to the user. Figure 6-11 shows a number of map visualizations opened for several attributes. Figure 6-12 shows how more visualizations are accommodated until capacity is reached.



**Figure 6-11: Choosing a map on the menu loads a map**



**Figure 6-12: Size and tiling are adjusted to fit more visualizations**

- **Table, not as “details first” but as “details on demand”:** There were a number of minuses outlined in the previous claims analysis with having the table of values being vertically aligned and “in-synch” with opening and collapsing tree-view folders. The table in the previous design was a “details-first” table. It shows more details than asked for; it attracts attention to tabulated details even before visualizations are opened. The new table acts as a “details-on-demand” table, after visualizations are loaded. The detail table is invoked by queries on the visualizations (details on demand). It shows values for only those attributes which are being viewed on the visualization. Table 6-4 summarizes important pros and cons for these features.

Design feature	Possible pros and cons
Right click on attributes to choose a corresponding visualization	+ Less actions required than dragging-and-dropping the attribute - but this action is not immediately obvious - but it may require a label directing users to do drag and drop
Label directing users to right click	+ reveals how to view a visualization - but may go unnoticed (banner blindness phenomenon)
Right click menu	+lists out all available visualization types - but users may not get a sense of what the visualization is, from the names on the menu
Ability to open several different kinds of visualizations, at will, all on same screen	+ Gives greater freedom and control - but novice users may not know what to do depending on tasks

**Table 6-4: Claims for design iteration-2**

An informal usability testing with three users revealed that the banner blindness phenomenon was indeed a problem. None of the three people noticed the label directing

them to right click. All of them tried clicking several times before they saw the label. Choosing visualizations according to task was also difficult. Understanding visual syntax and coordinations was still difficult, despite having a large histogram. The results were unsatisfactory. It was at this point that the idea for two additional hypotheses developed.

### 6.2.3 Final Design Iteration and Rationale

It was during the final design iteration when the bulk of the work of this thesis was done. The Learnable Visualization Elements Framework was clarified and two additional hypotheses were added. So the set of design hypotheses became:

- The “data-first” approach
- The “less visualization, more explanation tradeoff” approach
- The “predetermined task-based coordinations” approach

The rationale behind having the three principles together is this: combined, they address the three stages of the “visualization pipeline”:

Data → Visualizations → Coordinations

The data objects are the objects of interest. Data is what people know the. Data is mapped onto visualizations. This is the second most important thing to learn. Finally, once individual visualizations are understood, coordinating between visualizations reveals patterns and trends that could not have been found with single visualizations. Table 6-5 shows how different design elements were considered to make different items in the framework learnable.

<b>Learnable element</b>	<b>Instances of the learnable element in Datamaps</b>	<b>Design item that conveys the element</b>
Data Structure	<ul style="list-style-type: none"> <li>Data structure is hierarchical, attributes arranged under categories</li> </ul>	<ul style="list-style-type: none"> <li>Tree-view of attributes reveals hierarchy</li> </ul>
Data values	<ul style="list-style-type: none"> <li>All are numerical</li> </ul>	<ul style="list-style-type: none"> <li>Table reveals the nature of values immediately</li> </ul>
Data attributes	<ul style="list-style-type: none"> <li>All have a descriptive components plus the year in which it was measured</li> </ul>	<ul style="list-style-type: none"> <li>Tree-view of attributes – users get exposed to the nature of data attributes at start up. It is easy to see attributes like “population total 1997”, “population total 1998” and “Black population 1996”. It become clear from these examples.</li> </ul>
Data objects	<ul style="list-style-type: none"> <li>Data objects are states</li> </ul>	<ul style="list-style-type: none"> <li>The map is self explanatory that the items of interest are the states</li> <li>When the map coordinates along with a histogram bar, it clarifies that bars are states as well.</li> </ul>
Meta-data	<ul style="list-style-type: none"> <li>Units of measurement</li> </ul>	<ul style="list-style-type: none"> <li>Present in title of legend. Example: “Income in thousands of Dollars”</li> </ul>
Visual syntax	<ul style="list-style-type: none"> <li>Map color scale coding</li> <li>Histogram visual syntax</li> </ul>	<ul style="list-style-type: none"> <li>Legend (for map)</li> <li>Histogram is large and clearly labeled to clarify syntax: what is x-axis and what is y-axis</li> </ul>
Visual semantics	<ul style="list-style-type: none"> <li>Darker color is greater valued</li> </ul>	<ul style="list-style-type: none"> <li>Legends help understand this as well</li> </ul>
Operations	<ul style="list-style-type: none"> <li>Select states on map</li> <li>Deselect bars on histogram by dynamic query</li> </ul>	<ul style="list-style-type: none"> <li>The motivation of a user will make them want to try to operations. Specific design items were not used to reveal “all available operations”</li> </ul>

		<p>up front.</p> <ul style="list-style-type: none"> <li>• The dynamic query widget is indicative of an available operation on the histogram</li> </ul>
Operation mechanisms	<ul style="list-style-type: none"> <li>• Mouse click (to select a region on map)</li> <li>• Mouse click and drag(to select a group of states on map)</li> <li>• Dynamic query widget for dynamic querying</li> </ul>	<ul style="list-style-type: none"> <li>• Users will have to figure out mouse click by trial and error, in case of the map</li> <li>• Users will have to figure dragging to select multiple states by exploration</li> <li>• Mouse pointer image changes and tool tips were provided as cues to indicate how to move the dynamic query sliders</li> </ul>
Coordinations	<ul style="list-style-type: none"> <li>• Click region on map, view its value on table</li> <li>• Deselect bars on histogram, view deselected regions on map</li> <li>• Different coordinations help achieve different tasks</li> </ul>	<ul style="list-style-type: none"> <li>• The immediate feedback from the table was self explanatory for the map-table coordination</li> <li>• The dynamic legend help understand what the dynamic query sliders do to the map.</li> <li>• Having the map-table coordination on one tab, Map-histograms-and-dynamic-queries on a second tab, and multiple maps on a third tab helped implement hypothesis 2.</li> </ul>

**Table 6-5: Learnable Elements in Datamaps and design features that convey them**

The detailed description of design has already been given at the beginning of this chapter.

### 6.3 Summary

Overall, the final and new design of Datamaps was guided by the three hypotheses:

- Data related learnable elements were clarified by the tree-view. The tree-view first attracts attention toward the data (an implementation of hypothesis 1, the “data-first” approach).
- Visualization syntax and semantics related learnable elements were clarified using contextual explanations – large static or dynamic legends. These reveal the mapping between data and visualizations. (An implementation of hypothesis 2- the “more explanations” approach ).
- Coordination related learnable elements were primarily addressed by having different coordinations appear on different tabs, according to task. The labels on top of each tab, indicated what task the user could do in that tab. (an implementation of hypothesis 3- the “predetermined configurations” approach)



## 7 Usability Study

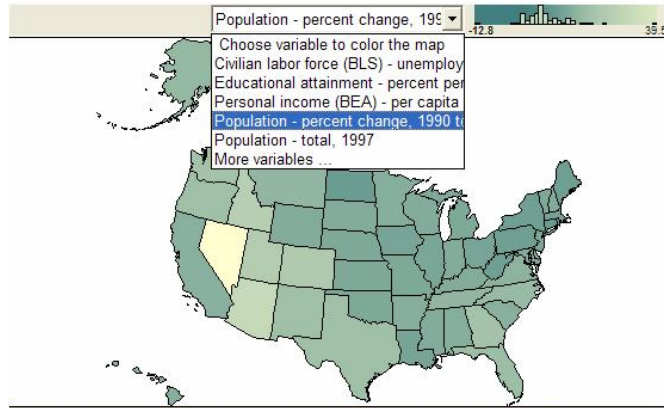
### 7.1 Objectives of the study

The study tries to answer the question: “Does adopting the three design hypotheses lead first-time novice users to adopt visualization-based strategies?” A version of Datamaps designed with the all three design principles was tested against an equivalent one (providing the same functionality) but designed without any of the three design principles. Both versions had alternate means of achieving tasks – without the use of visualization-based strategies. The metric used to measure understandability was the number of times users adopted visualization-based strategies in each interface while solving benchmark tasks.

### 7.2 Differences between the two versions of Datamaps

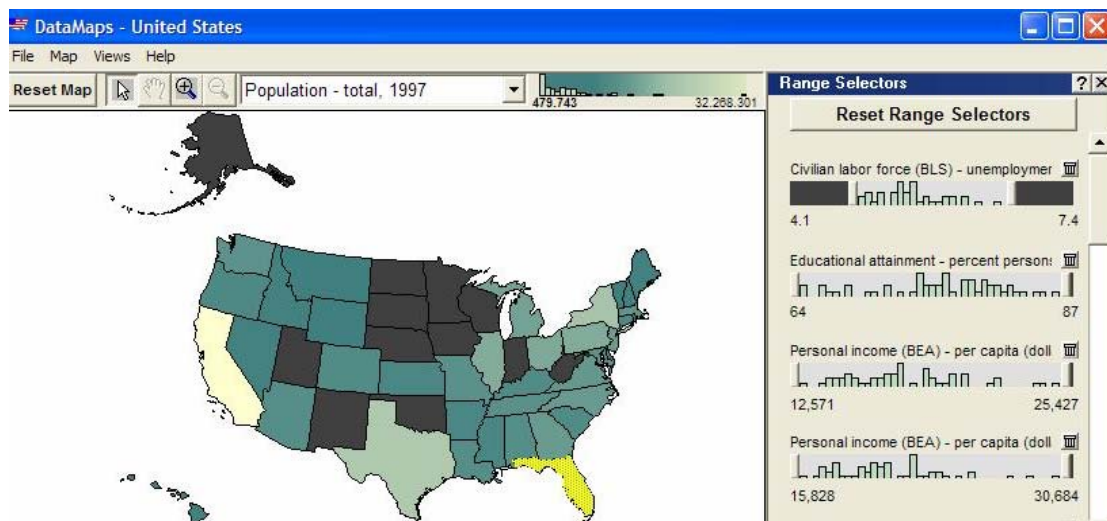
The new Datamaps does not include the scatter-plot view or the counties mode. In order to make the results comparable, these two functionalities were removed from the Old Datamaps. Both old and new designs of DataMaps were configured to have the following functionalities for experimental purposes:

- **Choropleth map:** Is always colored in new Datamaps; must be configured to become colored by an attribute in old Datamaps. Figure 7-1 shows how an attribute must be chosen from the drop down list, in order to configure the map to be colored by that attribute, in old Datamaps.

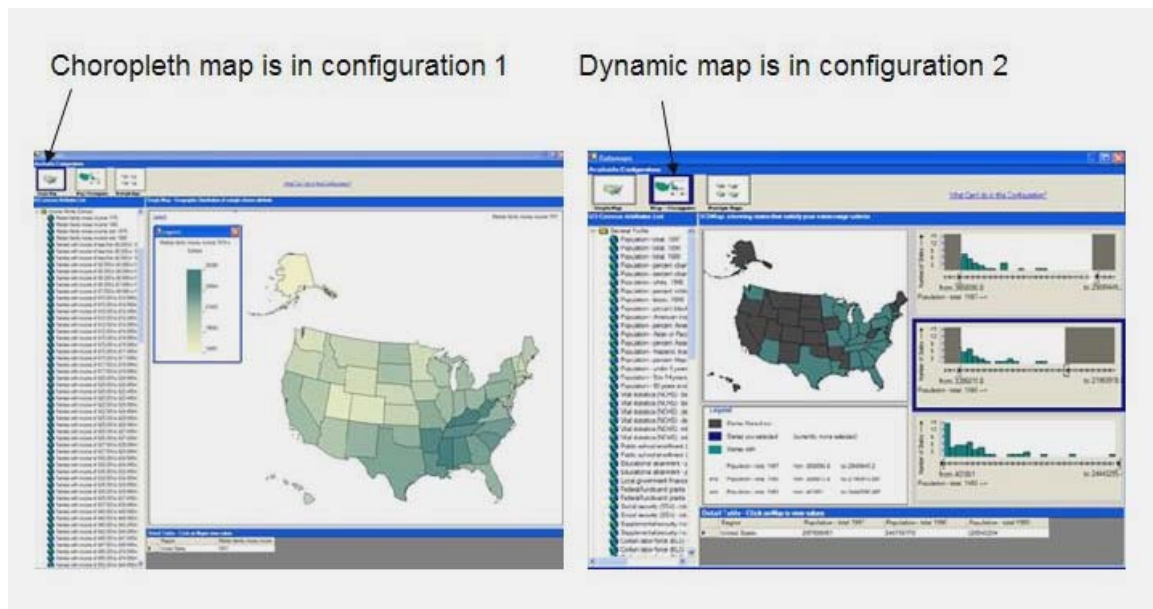


**Figure 7-1: Configuring the map to show color in old Datamaps**

- **Dynamic Map:** Is separate from the choropleth map in new Datamaps; the same map acts as dynamic query output and as choropleth map in old Datamaps. Figure 7-2 shows how the same map shows states colored by “Population total 1997” (while acting as choropleth map) and shows states being deselected due to a value range criterion being set for the attribute “Civilian labor force unemployment rate”



**Figure 7-2: Same map acts as dynamic map and choropleth map in old Datamaps**



**Figure 7-3: Choropleth map and Dynamic map on different tabs in new Datamaps**

- **Dynamic Query Sliders with histograms:** The histograms are larger, clearly labeled and calibrated in new Datamaps. The histograms are very small and y-axis has not been labeled to save space in old Datamaps.
- **Tree View of attribute list :** Prominently visible in new Datamaps; invoked by a button in old Datamaps
- **Detail Table:** The detail table displays details only for attributes that are currently being visualized in new Datamaps. The detail table displays details for all attributes from the current set of attributes, even the ones that are not being visualized.
- **Help:** There were just 3 help items in new Datamaps – a brief description of what each configuration does. The original version of Datamaps had 20 help items with sticky note help.

Table 7-1 shows the differences between Original Datamaps and New Datamaps in terms of how elements of the tool have to be learnt:

Activity	Original Datamaps	New Datamaps
Choosing an attribute	<ul style="list-style-type: none"> <li>Click on more variables button near histograms OR click on “more variables” option in map</li> <li>Scroll to required category</li> <li>Expand category</li> <li>Select a variable in the list</li> <li>Add it to favorites list by clicking the “&gt;&gt;” button (it is easy to miss this step, indicated by empirical data)</li> <li>Click OK</li> </ul>	<ul style="list-style-type: none"> <li>Scroll to required category</li> <li>Expand category (made easier to recognize with folder icons)</li> <li>Select a variable in the list</li> </ul>
Checking if the attribute has been chosen (added) in the interface	<ul style="list-style-type: none"> <li>Scroll the set of histograms</li> <li>OR Click on a state, to see table with an additional row (the newly added attribute). Scrolling the table may be necessary depending on how many rows there are</li> <li>OR check the drop down boxes against the map (this is not noticeable as empirical data indicates)</li> </ul>	<ul style="list-style-type: none"> <li>Notice the corresponding visualization that loads for the attribute</li> </ul>
Viewing a colored map	<ul style="list-style-type: none"> <li>Choose the attribute. Follow the steps for “choose attribute”.</li> <li>Choose the attribute from the drop down box against the map</li> </ul>	<ul style="list-style-type: none"> <li>Choose the attribute for the map. (The colored map immediately loads)</li> </ul>
Understanding the map color key		

	<ul style="list-style-type: none"> <li>• Look up legend. However, the legend is very small, not clearly labeled as “legend”.</li> <li>• OR Click states to verify the suspicion that color has meaning. (However, as empirical data indicates, users very rarely suspect that colors mean something despite noticing color changes)</li> </ul>	<ul style="list-style-type: none"> <li>• A clear legend is available upon clicking on a link labeled “legend”.</li> </ul>
Understanding the meaning of bar heights in histogram	<ul style="list-style-type: none"> <li>• Move the dynamic slider.</li> <li>• Notice states on the map changing colors</li> <li>• Deduce that one bar on the histogram stands for these states</li> </ul> <p>(empirical data indicates that users make several wrong deductions out of this coordination between map and histograms. Users making the correct deduction were never noticed in empirical data)</p>	<ul style="list-style-type: none"> <li>• Y-axis is labeled “number of states”</li> </ul>
Understanding the coordination between a single histogram and map	<ul style="list-style-type: none"> <li>• Notice states on the map changing colors and inferring the coordination.</li> </ul>	<ul style="list-style-type: none"> <li>• Legend is given</li> </ul>
Understanding the “logical AND” result on the map based on dynamic queries	<ul style="list-style-type: none"> <li>• Notice states changing color and infer the logic from it.</li> </ul>	<ul style="list-style-type: none"> <li>• An English sentence in the legend explains that green colored states are those that satisfy all criteria (the AND logic)</li> </ul>
Help	<ul style="list-style-type: none"> <li>• Help consists of several “sticky notes”- instructions about actions to perform on different parts of the interface. They are located on different parts of the interface.</li> <li>• Sticky notes often give confusing instructions that are unsuitable to the state of the interface. For e.g. sticky notes pointing towards an empty table.</li> </ul>	<ul style="list-style-type: none"> <li>• Help consists of a single small window with a picture of the configuration.</li> </ul>

**Table 7-1: Differences between the two tested versions of Datamaps**

## 7.3 Experiment Procedure

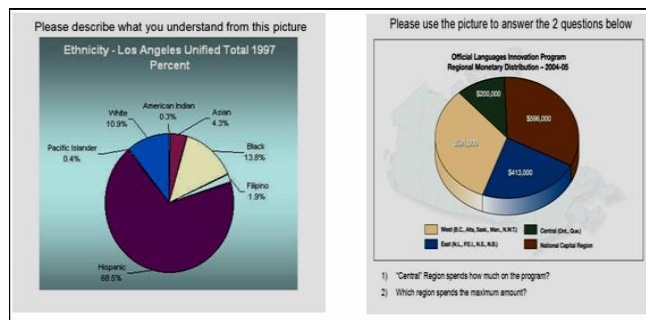
Eight participants were recruited in total. Several people from Blacksburg community were approached face-to-face in order to find subjects. Those who were suited criteria for recruitment were informed about the experiment procedures, benefits in research, duration of the experiment and that participation would be voluntary and uncompensated. Those who agreed to volunteer as subjects were randomly assigned to test on the old Datamaps or the new Datamaps. Let us call them Group A and Group B respectively. All participants of both groups:

- Were adults above age eighteen
- Did not have previous experience with interactive visualizations
- Had graphical-user interface experience from one or more of: email, browsers, instant messaging software, multi-media playing/editing software.
- Were frequent users of computers and internet
- Had used computers for several years
- Had at least a vague sense of what “Census” meant
- Had enough familiarity with the US map to locate at least a few states

Group A and Group B were also checked for balance in composition of group members by their backgrounds. Each group had:

- Two subjects with design oriented backgrounds: Graphic design, engineering, computer science
- Two subjects without design oriented backgrounds: Medicine, Food service, Elementary school teaching.

All participants were required to sign an informed consent document upon arrival for experiment (included in Appendix E). All participants filled up a demographics questionnaire (Appendix A), and also took a screening test (Appendix B) for color blindness test and a graphical reading ability test. The graphical reading ability test had a pie chart of racial group percentages in Los Angeles. To determine if participants could make meaningful interpretations, they were asked to describe what they understood. A second pie chart required them to look up a legend to identify what the pies were – this was slightly more attention demanding than the previous question due to the look-up. Figure 7-4 shows the two charts used for the graphical reading ability tests.



**Figure 7-4: Pie charts to test graphical reading ability of participants**

Participants then proceeded to the main experiment with a task list (Appendix C). They completed a “pre-task session” in which they familiarized themselves with the Data-maps software. They were asked to think aloud during this session. After this session, they were asked to perform tasks 1 to 9, while thinking aloud. The screen, their faces and voices were captured with TechSmith Morae software. After the session, they completed an exit questionnaire (Appendix D) which determined whether they were already familiar with visualization tools and also obtained information about their graphical user interface familiarity. Tables 7-2 and 7-3 show demographics collected

from Group A (old Datamaps group) and Group B (new Datamaps group) participants respectively.

Item	User A1	User A2	User A3	User A4
Age	23	62	34	23
Gender	M	M	F	F
Occupation	Student	Doctor	Dentist	Shift Manager
Major/Background	Business, Computer Science	Medicine	General Dentistry	Marketing Graphics Design
Computer usage frequency	Every day	Every day	Every day	Every day
Computer total usage	several years	several years	several years	Several years
Softwares used	MS-Word Excel .Net Outlook MS-Paint Powerpoint FrontPage RealPlayer Win Media Player Fruityloops(music production)	Internet Explorer Yahoo messenger Yahoo mail MS Word	MS Word Flash Acrobat Reader Yahoo Messenger Outlook	MS Access MS Word MS Excel MSPublisher MS Outlook Dreamweaver Fireworks
Familiarity with US state names	Know several state names	Know several state names	Know all state names	Know all state names
Familiarity with locating states on map	Can locate several	Can locate several	Can locate several	Can locate several
Come across Census data about USA?	Yes	Yes	Yes	No
Where?	TV, Internet, Class	TV	TV, Internet	----
Familiarity with census data	“Have vague sense of what kinds of items Census data is collected for”	“Have vague sense of what kinds of items Census data is collected for”	“I sometimes use Census data for my knowledge/work”	“I know what Census means, nothing more”
Colorblindness?	No	No	No	No
Graphical reading ability questions	Passed	Passed	Passed	Passed

**Table 7-2: Group A demographics**



Item	User B1	User B2	User B3	User B4
Age	29	24	22	37
Gender	M	M	M	F
Occupation	Restaurant Management	Student	Student	Home-school teacher
Major/Background	Food Service	Civil Engineering	Electrical Engineering	Elementary Education
Computer usage frequency	Few times a week	Every day	Every day	Every day
Computer total usage	several years	several years	several years	Several years
Softwares used	Yahoo Mail Netscape mail Lycos mail HP CD/DVD Burner Windows Media Player	MS Office MS Project Intuit Master Builder AutoCAD Heavybid express	MS Office Yahoo Mail Hotmail GMail Adobe Lotus Foxpro Basic C/C++ Matlab Flash	Apple email client Firefox web client MS Word MS Excel Powerpoint PageMaker Lotus Notes Only Mac Familiarity No windows familiarity
Familiarity with US state names	Know all state names	Know a few state names	Know several state names	Know most state names
Familiarity with locating states on map	Can locate almost all states	Can locate a few states	Can locate a few states	Can locate several states
Come across Census data about USA?	yes	No	No	Yes
Where?	TV Internet	---	---	Newspapers Internet
Familiarity with census data	“Have vague sense of what kinds of items Census data is collected for”	“Have vague sense of what kinds of items Census data is collected for”	“Have vague sense of what kinds of items Census data is collected for”	“Have vague sense of what kinds of items Census data is collected for”
Colorblindness?	No	No	No	No
Graphical reading ability questions	Passed	Passed	Passed	Passed

**Table 7-3: Group B demographics**

Table 7-4 presents the task list that the participants were given. The question type and the recommended strategy to solve the question are tabulated. Visualization based strategies are the recommended strategies. Questions (1) and (6) are “point-value” questions asking for a specific number – these two cannot be solved visually.

	QUESTION	QUESTION TYPE:		RECOMMENDED STRATEGIES
		GIVEN	FIND	
1	What was the population of California in 1997?	An attribute, a state	the value for that state	Read individually
2	Compare the “Personal income [BEA] – per capita [dollars], 1994” values for any 2 states not next to each other. Which had greater per capita income?	An attribute, value criterion, a sub-set of states	the state with max value	By color, read individually
3	Of the three states on the west coast, which state had the lowest civilian labor force unemployment rate in 1996?	An attribute, value criterion, a sub-set of states	the state with min value	By color, Read individually
4	Name the states which had less than 50% white population in 1996. Refer to “Population - percent white, 1996” while you work at this question.	An attribute, value criterion	the states that satisfies criterion	By DQ, color
5	Which two states had the highest population percent change from 1990 to 1997? Refer to “Population – percent change, 1990 to 1997” while you work at this question.	An attribute, value criterion	the states that satisfy criterion	By DQ, color
6	How many families in Minnesota lived below the poverty line in 1989? Refer to “Families below poverty level in 1989” while you work at this question. This is found under “Poverty” category.	An attribute, state	the data value for that state	Read Individually
7	Find a state that has a) LOW number of “marriages per 1000 population” in 1988 AND LOW number of “divorces per 1000 population” in 1988. Note: Marriages and Divorces data are found under “Vital statistics- Births, Deaths, Marriages, Divorces”	Two attributes, value range criteria	the states that satisfy both criteria	DQ, By color
	b) HIGH number of “marriages per 1000 population 1988” AND HIGH number of “divorces per 1000 population 1988”	Two attributes, value range criteria	the states that satisfy both criteria	DQ, By color
	c) LOW number of “marriages per 1000 population 1988” AND HIGH number of “divorces per 1000 population”.	Two attributes, value range criteria	the states that satisfy both criteria	DQ, By color
8	Which state in the USA had the highest population in 1997?	An attribute, value criterion	the state with max value	By color, DQ
9	Point to clusters of states on the map having low values for educational attainment. Refer to “educational attainment – percent persons 25 years and over, high school graduates, 1990” under “general profile” while you work at this question.	An attribute, value criterion	the states that satisfy value range criteria	By color, DQ

**Table 7-4: Task list for usability study on new Datamaps**

## 7.4 The two visualization-based strategies possible in the experiment

As seen in table 7-4, questions can be solved by reading tables or with either of the two visualization-based strategies: using map color or using dynamic querying. **Using map color to determine trends is a visualization-based strategy because it involves:**

- **Understanding visual syntax:** that map color scale stands for numerical values
- **Interacting with a visual representation:** That is, clicking on states to view specific values

**Using map color helps identify values for particular states, and states with extreme values.**

The other strategy is using dynamic queries. **Dynamic querying is also a visualization-based strategy because it involves:**

- **Understanding visual syntax**– Users have to understand that bars stand for number of states having particular value ranges in the histogram.
- **Interacting with visual representations** – Once they understand that bars represent number of states, they may use the sliders to interactively set value ranges and observe the states that satisfy the value range criteria.
- **Another round of understanding visual syntax:** They also have to understand that graying out states on the map mean that these states are getting filtered out.

## **7.5 Rationale for recommending visualization-based strategies**

In both versions of Datamaps, tasks 2, 3, 4, 5, 7a, 7b, 7c, 8 and 9 can be completed by using either a visualization-based strategy or textual-representation based strategy (that is by reading tables of values). Tasks solved by using visualization-based strategy can be achieved much faster. For some tasks, reading tables until an answer is obtained might be so laborious that users may want to quit. This is why any of the visualization-based strategies are listed as recommended strategies (as opposed to strategies based on reading tables).

## **7.6 Performance of participants – quantitative results**

Table 7-5 presents the strategies used by participants while they attempted to solve the tasks. “Not attempted” entries mean the question was not presented to them due to time constraints or technical glitches, or they chose to skip the question. 7 a, b and c were sufficiently similar and hence participants were allowed to approach them in any order, or skip one or two of them. The gray boxes indicate that a visualization-based strategy (either colored map or dynamic querying) was used rather than just reading off values from the table.

	A1	A2	A3	A4	B1	B2	B3	B4
1	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually
2	Read values individually	Read values individually	Read values individually	Read values individually	Read values individually	Read values individually	Read values individually	Read values individually
3	Read values individually	Read values individually	Read values individually	Read values in groups	By color	Read values individually	Read values in groups	By color
4	Dynamic querying	Read values individually	Read values individually	Read values in groups	By color	Dynamic querying	Read values in groups	By color
5	Dynamic querying	By color	Read values individually Quit in a while	Read values in groups	By color	Dynamic querying	Read values in groups	By color
6	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually	Read value individually
7 a	Not attempted	By color	Not attempted	Not attempted	By color	Dynamic querying	By color	By color
7 b	Dynamic querying	By color	Not attempted	Not attempted	By color	Dynamic querying	By color	By color
7 c	Not attempted	Read values by group – Incomplete	Read values Individually. Quit in a while.	Read values in groups	Dynamic Querying	Dynamic querying	Dynamic querying	Not attempted
8	Dynamic querying	Read values individually	Read values in groups	Not attempted	By color	Dynamic querying	By color	By color
9	Dynamic querying	By color	Not attempted	Read values in groups	By color	Dynamic querying	By color	By color

**Table 7-5: Strategies of Group A (old version) & Group B (new version) participants. Gray cells indicate that a visualization-based strategy was used. Visualization-based strategies are recommended strategies. More gray cells are found under the New Datamaps, indicative of greater success.**

“Read values individually” implies the participant used the strategy of clicking on 50 different states one by one and reading values from the table. Only one state is loaded at a time, in the table. “Reading values in groups” implies the participant used a rectangular selection to select several states at once, and scrolled the table to read values

– this involves a lesser amount of mental book-keeping, since several states are visible on the table at once. “Dynamic querying” implies they moved sliders to arrive at the answer (it does not include whether or not they understood the bar histograms- this issue is addressed in a separate table). “By color” implies they used the map color to arrive at the answer. Out of these, “by DQ” and “by color” are visualization-based strategies – since they involve understanding and using colors or color coordinations rather than just reading tables. Tables 7-6 and 7-7 summarize usage of visualization-based strategies in Group A and B:

Task	Number of people out of four using visualization-based strategies	
	In Old Datamaps (Group A)	In New Datamaps (Group B)
1	---not applicable---	---not applicable---
2	0/4	0/4
3	0/4	2/4
4	1/4	3/4
5	2/4	3/4
6	---not applicable---	---not applicable---
7	2/4	4/4
8	1/4	4/4
9	2/4	4/4

**Table 7-6: Number of people using visual strategies**

Visualization-based Strategy	Total number of times strategy was used to solve questions	
	In Old Datamaps (group A)	In New Datamaps (group B)
Using Map color	4	18
Using DQ sliders	5	9
Totally visualization-based strategies used	9	27

**Table 7-7: Number of times visualization-based strategies were used**

## 7.7 Performance of participants – qualitative results

Task 1, the simplest task, can be answered only by clicking the required state on the map and reading the value table. Going from 1 to 9, it can be seen that users seem to have a tendency to latch on to a strategy in successive questions:

- Task 1 can be solved by reading the table only. This seems to have influenced every user (including those who knew the color syntax from pre-task session) to do task 2 by clicking and reading, while it could have been easily solved by looking at state colors.
- B1 discovered the map color syntax during the pre-task itself, by clicking on the legend. She is seen using the map colors right from task 3 to task 9. She discovered and understood the use of histograms for question 7c.
- B4 also discovered the map color syntax during the pre-task itself, by clicking on the legend. He too, is seen using the map colors right from task 3 to task 9. He avoided the use of histograms, because he chose to quit question 7c – which is the only question that absolutely requires the use of DQ's to answer efficiently.
- B2 was frustrated with having to click 50 states in task 4. So he explored for other options and discovered that DQ's solve the problem efficiently. He can be seen using the DQ strategy from task 4 to task 9. He completely understood histogram syntax and relied on dynamic querying alone. He did not click on the legend, so he never used map color in solving questions. After all questions were over, he clicked on the map legend and understood map colors.
- B3 used the histogram and slider correctly during the pre-task itself. However, he seems to have been influenced by task 1 so much that he used selection and reading from table until task 7. He used both DQ and color to solve questions 7a,

- 7b and 7c. Armed with the newly discovered “use map color to detect trends” strategy, he solves questions 8 and 9 by map colors in a matter of 2 seconds each.
- A1 didn’t understand the histogram syntax, but he managed to use sliders to arrive at answers and one can see him using DQ starting from question 4 onwards. He did not understand the histogram bars. He took the deselection (darkening of states on map while query sliders are moved) to mean selection.
  - A2 used selecting states on the map and reading from the table. During task 5, it occurred to A2 that the colors of the map might mean something. From then on, he mainly uses color to answer questions. He did not understand histograms or sliders.
  - A3 used selecting states on the map and reading from the table. She did not discover the color syntax and did not understand histogram syntax.
  - A4 used selecting states on the map and reading from the table. She did not discover the color syntax and did not understand histogram syntax.

## **7.8 Discussion of usefulness of new design approaches**

### **7.8.1 The data-first approach**

As in the previous study by Census department, **users still tried to “match” chunks of information needs to visible elements on screen that seemed to help with those needs.** In the old Datamaps the most prominent chunks at start up were the data objects – the states.

But in the new Datamaps, **since the data is displayed prominently upfront, the visible elements on the screen (both map and data tree-view) now helped with both kinds of information needs:**



- States (Data objects) – The map shows clickable states, which are the data objects.
- And Attributes (Data attributes) – The tree-view shows clickable attributes of states which are the data attributes.

**Having the data displayed prominently achieved the following things:**

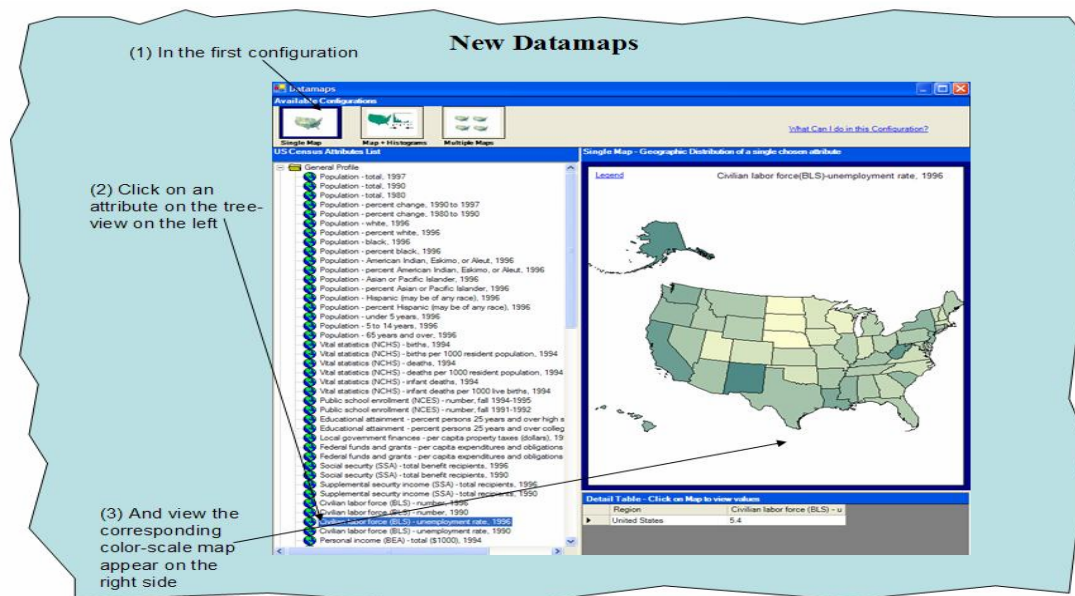
- **Understanding of the data structure:** Users had an immediate understanding of the structure of the data, by looking at the tree-view. This was evident by remarks such as “Let me see what attributes are there in the crime category” during the pre-task session. It was also evident by the ease with which users scrolled through and navigated the tree-view. To find particular attributes, they would go to related categories and open them.
- **Understanding of the nature of the data attributes:** Since the “General Profile” category is kept open by default in the tree-view, attributes such as “population total, 1997”, “population total, 1990” were prominently visible. This revealed the nature of the attributes – each has a descriptive component and a year component.

“Provide a mechanism to input data attributes into visualization components with minimal user actions” – this is one of the aspects of the Data-first hypothesis. The way this has been implemented is “Simply click on an attribute, and the corresponding visualization appears”. **The “single click on attribute to view visualization” interaction design achieved the following things:**

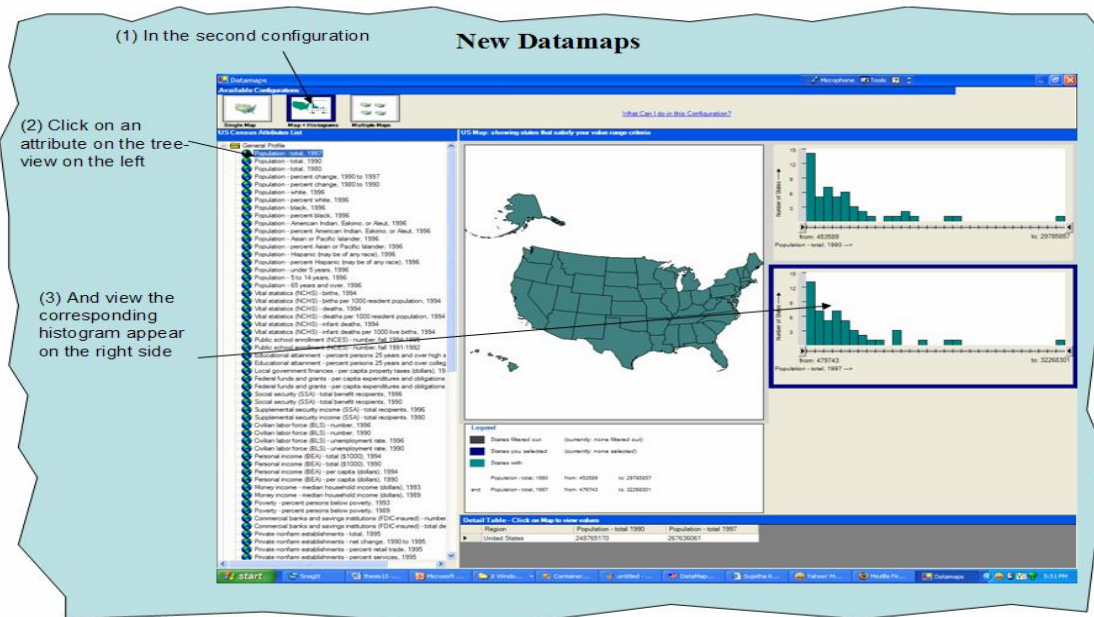
- **Understanding the relationship between data attributes and visualizations:** Users quickly associated the two. This interaction design

made it clear to all four users of the new Datamaps that “every attribute has a visualization.” This was evident by remarks such as: “Now I see [the map for population] 1997. I need [the map for population] 1990. So [let me] click on [the attribute population] 1990.”

- **Organizing a left-to-right action path (data attribute first, visualization next):** The placement of the tree view on the left along with the single click interaction design helped bring about a clear and simple action-path. “Click on attribute and view visualization”. Figures 7-5,7-6 and 7-7 illustrate the action path.



**Figure 7-5: Left-to-right action path in configuration 1, New Datamaps**



Left-to-right action path in configuration 2, New Datamaps

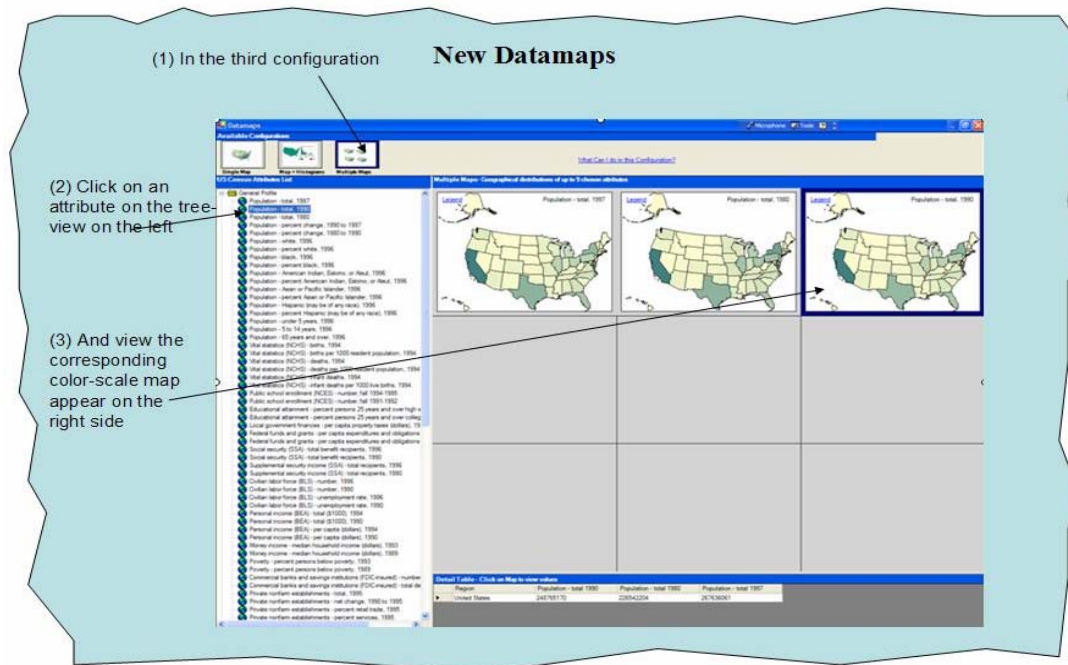


Figure 7-6: Left-to-right action path in configuration 2, New Datamaps

By contrast, in the old Datamaps, users took a longer time to realize the action-path that when the required attribute was not visible, they have to click on “more variables” and follow an “add attribute” procedure.

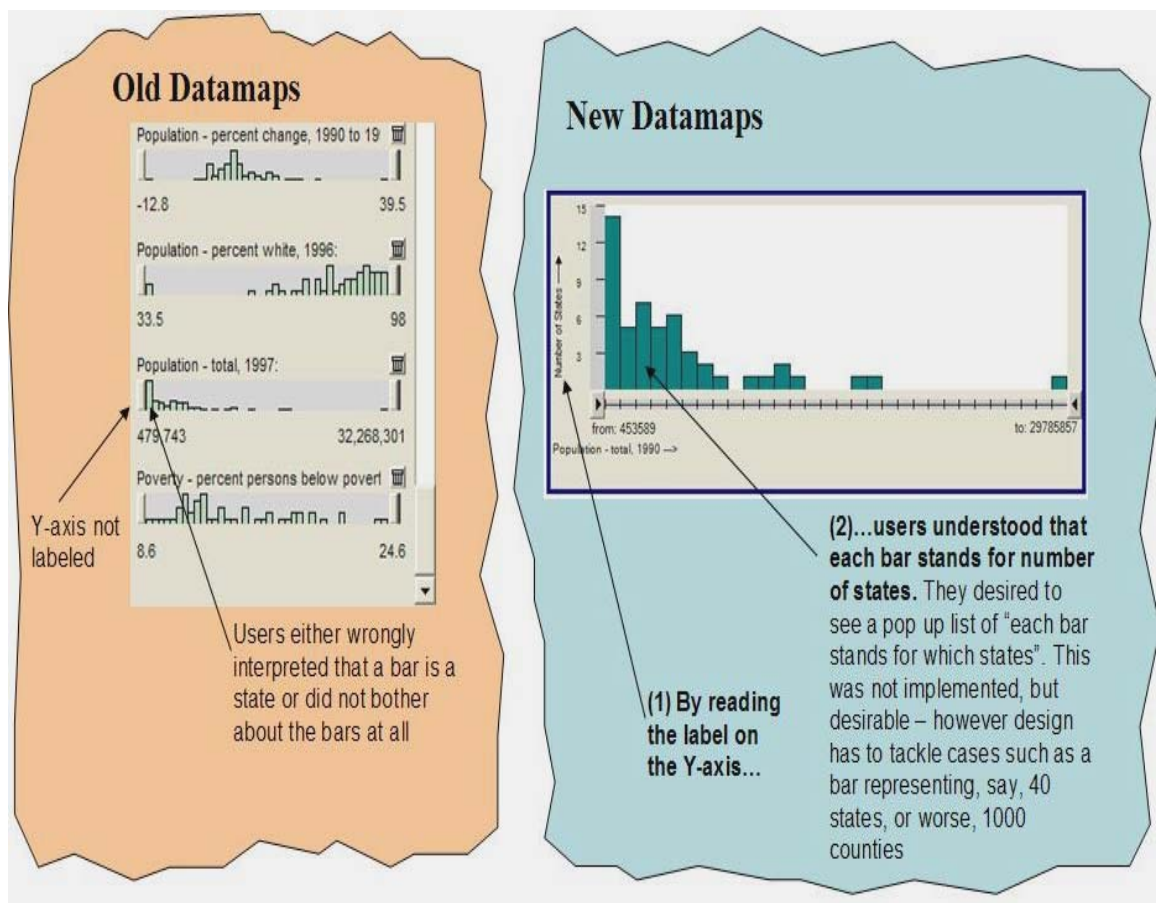
An interesting point to note is that when users scrolled through the tree-view in the old Datamaps, the data structure and the nature of the attributes became clear to them. Again, this shows that having a data-first approach helps in understanding the data, and that **understanding the data is important prerequisite to being able to use visualization-based strategies effectively**. Understanding the data helps navigate the whole data set to find required information. A **simple and quick interaction design, to view visualizations for given attributes** helps in understanding the relationship between attributes and visualizations. It **eliminated the configuration step which was a hindrance** in understanding this.

Users suggested that having an alphabetical ordering of attributes would make it easier to navigate. Users said they would have no problem scrolling through even a long list of attributes, if only they were alphabetical – because they are used to doing alphabetical scanning with their eyes.

### **7.8.2 “Less visualization for more explanation approach”**

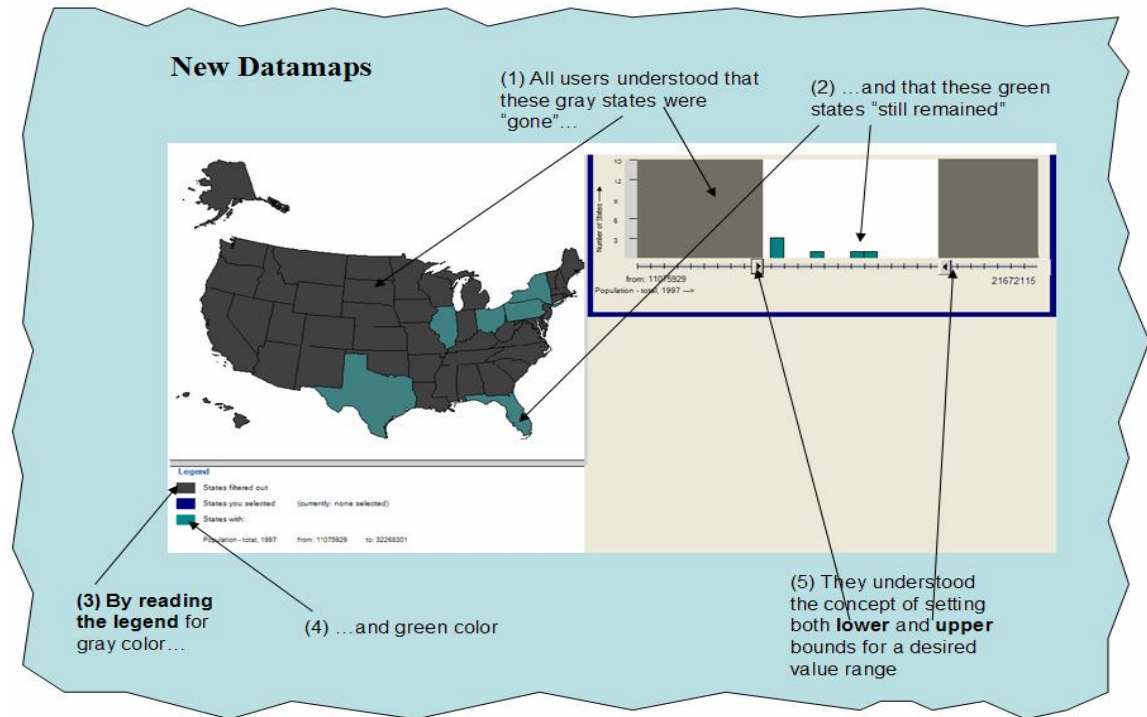
This approach stresses the importance of having clear explanations for the visual syntax. The **labels and legends were of great help in understanding visual syntax**. Figures 7-7, 7-8 and 7-9 illustrate the benefit of labels and legends. **All four users in new Datamaps understood the visualization strategy based on map color by reading the color legend**. One user did not click the legend link in new Datamaps, but he too, said “I would have definitely used the legend to answer several questions had I clicked on it

before.” This was the user who relied mostly on dynamic querying to solve the questions. Now, it was desired during design that the legend be visible prominently without the need of a legend link. But due to an implementation issue, the link was used. Three users in the old Datamaps never even thought about what the color scale on the map means. One user discovered color meaning by educated guesses. Based on all the evidence discussed, **it is fair to say that making room for explanations on the screen is very important to lead novices to use the visualization-based strategy.**

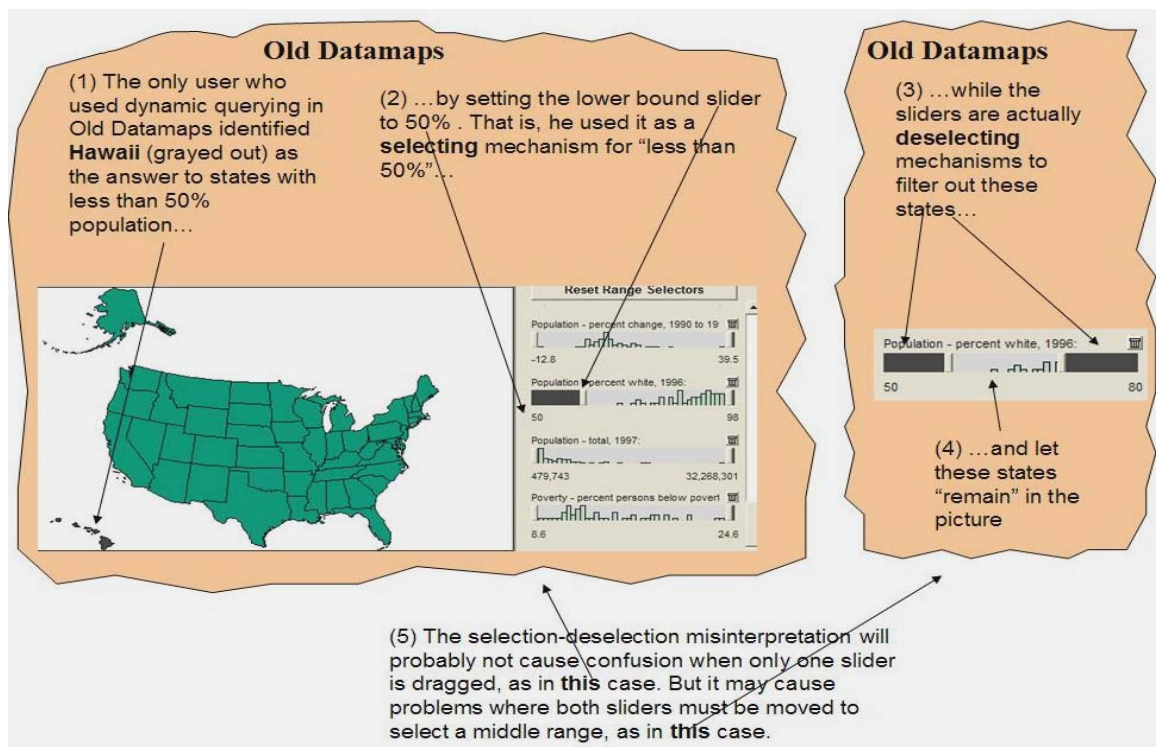


**Figure 7-7: Labels help in understanding visual syntax**





**Figure 7-8: Legend helps in understanding visual syntax**



**Figure 7-9: Selection-Deselection confusion without the legend in Old Datamaps**

### 7.8.3 Predetermined Configurations approach

The predetermined configuration approach was designed to expose users to increasing complexity. It was also designed to give ready-made configurations that help achieve specific tasks. Did it succeed? Evidence from the experiment indicates that it did succeed in achieving these:

- **Learning incrementally:** All four users were able to learn the on the single map configuration – that is the map table coordination. Even while looking at the navigation section, they were able to predict how this would extend to the multiple map configuration. When they switched to the multiple map configuration from the single map configuration, all they had to learn was how to add additional maps on to the screen. Once they were added, all the maps behaved in the same way.
- **Switching between configurations depending on task:** All four users were able to make sensible switches between configurations. For example, if they were on the single map configuration, and the task could be achieved by comparing two maps, they would switch to the multiple-map configuration. In another case, a user observed: “Now I am on single map configuration. I want to know states with less than 50%. Perhaps I need to go to the...uh... map plus histograms configuration now”. He switched over, used dynamic querying and succeeded in the task. He had already learnt dynamic querying and all the configurations during the pre-task.

## **7.9 Tradeoffs involved in designing for novices**

The data-first approach shows more of the data structure; this takes space away from showing more visualizations.

The predetermined task-based coordinations approach also shows particular configurations at a time. This means, not being able to show all available visualizations at once. This relies on the exploring nature of users. Users are expected to switch between available task-based configurations, using a navigation bar with icons. Clicking on each icon should lead to a configuration. However, the experiment suggests that this was not a problem – all users explored all the options. They even gave rationale for switching between configurations, when one configuration seemed unsuitable for what they needed to do.

The “less visualization, more explanation tradeoff” approach, indicates the tradeoff in the name itself. For novices, it may be more important to tell them whatever it is they are seeing – even if that means less visualization, than giving them “too much of everything”, which could lead to “information overload”.



## 8 Conclusions and Future Work

### 8.1 Conclusions

In the beginning, this work posed the question: “**Can interface design enable first-time novice users of a coordinated-view visualization tool to employ visualization-based strategies to gain insights from data? If so, how?**” The answer to that question is “Yes, through effective design principles, it is possible to lead novice users to use visualization-based strategies”.

This work achieved the answers by deriving design principles based on a framework of “Learnable Visualization Elements” in a relational-data visualization tool with coordinated-views. The framework is hierarchical and consists of the following items:

- Data
  - Data structure
  - Data attributes
  - Data tuples
  - Data values
  - Meta-data
- Visualization
  - Syntax
  - Semantics
  - Operations
  - Operation executing mechanisms or Interaction mechanisms
- Coordination
- Overall interface-level controls

The learnable elements framework outlines certain prerequisites for being able to use visualization-based strategies. Especially important prerequisites are:

- An understanding of data attributes and data structure: This is prerequisite for being able to navigate a large set of data attributes. The data-first design approach helped achieve this.
- An understanding of visual syntax: This is a prerequisite for being able to use visual representation.
- The ability to choose coordinations according to task: This is necessary to make effective use of coordinated visual mappings. The predetermined configurations approach helped users navigate between suitable coordinations depending on the task.

The design approaches were implemented in a multi-dimensional coordinated-view visualization tool called Datamaps. A new version of Datamaps designed based on all three design principles was tested against an equivalent version (providing similar functionality) designed previously. Both versions had alternate means of achieving tasks – without the use of visualization-based strategies. The objective was to examine if the three design principles causes novices to adopt visualization-based strategies instead of using the alternate means of reading tables. The metric used to measure understandability was the number of times users adopted visualization-based strategies in solving nine benchmark tasks. The new Datamaps succeeded in enabling novices to use visualization-based strategies.

Hence the three design approaches: **“Data-first”**, **“Less visualization for more explanation”** and **“predetermined task-based coordinations”** are found to be useful

**design principles** in the context of coordinated-view relational data visualizations for first-time novices.

## **8.2 Potential implications**

There may exist visualizations that already employ the design approaches derived here. But the contribution of this thesis is that it discovers and documents these as useful design principles; it examines the details behind the successes of those visualizations that do use these design principles.

This has been preliminary work in designing for first-time novices. Much more research is necessary in this area to form a large pattern language for visualization design. The term “pattern language” coined by Christopher Alexander and popularized in his book “A Pattern Language: Towns, Buildings, Construction” (Alexander 1977) is used to mean a structured method of describing good design practices within a particular domain.

Research in this area is hoped to increase the use of interactive visualization-strong interfaces (discussed in the chapter titled “Background”). For example, in domains such as digital libraries, emails and search engines, there is still widespread use of textual and tabular presentation of results. Easy-to-use interactive visualizations for novices may easily be designed for these applications.

## **8.3 Future work**

The three design approaches: the “Data-first” approach, the “predetermined task-based coordinations” approach and the “less visualization more explanation tradeoff” approach show promise. However these design hypotheses are based on the three largest elements in the “Learnable Visualization Elements” framework.

Finer design hypotheses, made for the finer elements in the framework hierarchy may be tested in the future. For example, consider the element “operations” under “Visualizations”. All operations allowed on a particular visualization may be presented as buttons with icons as seen in toolbars and tested against, say, operations which are hidden within menus. This level of analysis was not done in this work.

Another interesting area would be to test multiple attribute visualization types like scatter-plots or parallel plots. This work deals with single attribute visualizations like choropleth maps and histograms. So the interaction design was simple: simply click on an attribute and load the visualization for that. Getting the data attribute from the tree-view browser to the visualization was easy. When dealing with double attribute visualizations like scatter-plot similar simple interaction design is necessary. More research is necessary for finding quick and simple ways to get data from a data-exploration view for greater-dimensional visualization types.

Other important considerations are how many predetermined configurations are necessary? How can we make users navigate between them? Can we include the option to construct custom configurations, after expertise is acquired? In this way, the tool could serve both novices and experts at the same time.

Several information visualization tools exist that are design for applications whose target users are not expected to be domain experts. For example, search engine interfaces are in the domain of the Internet. Few people possess the knowledge about the content or structure of all web sites in the world. There are advanced visualizations being designed every day for such domains whose users have little knowledge about the domain. These can all be tested with data-first design, the task based configurations design. This would

help measure the applicability of these design principles to a wide variety of visualization tools.

A pattern language for design of visualizations may be generated, eventually. Broad guidelines are difficult to adapt to specific situations. Pattern-languages constructed after several empirical studies in several contexts may prove more useful to designers than broad guidelines. This work has been one step in this direction.

## 9 References

- Ahlberg, C. (1996). "Spotfire: an information exploration environment." ACM SIGMOD Record archive **25**(4): 25-29.
- Alexander, C., Ishikawa, S. and Silverstein, M. (1977). A Pattern Language: Towns, Buildings, Construction, Oxford University Press.
- Carroll, J. M., Rosson, M.B. (1987). The paradox of the active user. Cambridge, MA, MIT Press.
- Cleveland, W., McGill, R. (1984). "Graphical perception: theory, experimentation, and application to the development of graphical methods." Journal of the American Statistical Association **79**(387): 531-554.
- Cleveland, W., McGill, R. (1985). "Graphical perception and graphical methods for analyzing scientific data." Science **229**(August): 828-833.
- Draper, S. (1986). Display Managers as the Basis for User-Machine Communication. User Centered System Design Lawrence Erlbaum Associates. D. a. D. Norman, S.
- Dyck, J. L. (1995). "Problem solving by novice Macintosh users: the effects of animated, selfpaced written, and no instruction." Journal of Educational Computing Research **12**: 29-49.
- Eklund, P. W., Ducrou, J. and Brawn, P. (2004). Information Visualization using Concept Lattices: Can Novices Read Line Diagrams? 2nd International Conference on Formal Concept Analysis, Springer-Verlag.
- Harrison, S. M. (1995). A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface. ACM Conference on Human Factors in Computing Systems, Denver, CO.
- Howes, A., Payne, S.J. (1990). "Display-based competence: Towards user models for menu-driven interfaces." International Journal of Man-Machine Studies **33**(6): 637-655.
- Howes, A., Young, R.M. (1991). Predicting the learnability of task-action mappings. Proceedings of the ACM CHI '91 Human Factors in Computing Systems Conference, New Orleans, Louisiana.
- Kang, H., Plaisant, C., and Shneiderman, B. (2003). New Approaches to Help Users Get Started with Visual Interfaces: Multi-Layered Interfaces and Integrated Initial Guidance. National Conference on Digital Government Research.
- Kennedy, J. B., Mitchell, K.J. and Barclay, P.J. (1996). "A Framework for Information Visualisation." SIGMOD Record **25**(4): 30-34.
- Kitajima, M. P., P.G (1992). A computational model of skilled use of a graphical user interface. Proceedings of the ACM CHI'92 Conference on Human Factors in Computing Systems, New York.
- Koffa, K. (1935). Principles of Gestalt Psychology. London, Routledge and Kegan Paul.
- Lakkaraju, K., Yurcik, W., Lee, A.J. (2004). NVisionIP: netflow visualizations of system state for security situational awareness. Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, Washington DC, USA.
- Mackinlay, J. (1986). "Automating the design of graphical presentations of relational information." Transactions on Graphics **5**(2): 110-141.

- Marks, L., Hussell, J., McMahon, T.M, Luce, R. (2005). "ActiveGraph: A digital library visualization tool." International Journal on Digital Libraries **5**(1): 57-69.
- Mayes, J. T., Draper, S.W., McGregor, M.A. and Oatley, K. (1988). Information flow in a user interface: the effect of experience and context on the recall of MacWrite screens. People and Computers IV, Cambridge University Press.
- McGrenere, J., Baecker, R. and Booth, K. (2002). An evaluation of a multiple interface design solution for bloated software. Proceedings of CHI'02 Human Factors in Computing Systems.
- Neerincx, M., De Greef, P. (1993). How To Aid Non-Experts. Conference on Human Factors in Computing Systems.
- Norman, D. (1991). Cognitive artifacts. Designing Interaction: Psychology at the Human Computer Interface. J. Carroll. New York, Cambridge University Press.
- North, C. (2000). A User Interface for Coordinating Visualizations Based on Relational Schemata: Snap-Together Visualization. College Park, University of Maryland. **Ph.D.**
- Payne, S. J. (1991). "Display-based action at the user interface." International Journal of Man Machine Studies **35**: 275-289.
- Payne, S. J., Chesworth, L. and Hill, E. (1992). "Animated demonstrations for exploratory learners." Interacting with Computers **4**: 3-22.
- Pfiftzner, D., Hobbs, V. and Powers, D. (2003). A unified taxonomic framework for information visualization. Proceedings of the Australian symposium on Information visualisation, Adelaide, Australia.
- Plaisant, C. (2004). The challenge of information visualization evaluation. Proceedings of the working conference on Advanced visual interfaces, Gallipoli, Italy.
- Rieman, J. (1996). A field study of exploratory learning strategies. ACM Transactions on Computer-Human Interaction (TOCHI).
- Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Proceedings of the IEEE Conference on Visual Languages, Boulder, CO.
- Shneiderman, B. (1998). Designing the User Interface: Strategies for Effective Human-Computer Interaction (third edition). Reading, MA, Addison-Wesley.
- Shneiderman, B., Kang, H., Kules, B., Plaisant, C., Rose, A. and Ruchair, R. (2002). "Design: A photo history of SIGCHI, evolution of design from personal to public." Interactions **9**(3): 17-23.
- Tufte, E. R. (1990). Envisioning Information. Cheshire, CT, Graphics Press.
- Tversky, B., Morrison, J.B. and Betrancourt, M. (2002). "Animation: can it facilitate?" International Journal of Human-Computer Studies **57**(4): 247-262.
- Tweedie, L. (1997). Characterizing interactive externalizations. Proceedings of the SIGCHI conference on Human factors in computing systems.
- Williamson, C. a. S., B. (1992). The Dynamic HomeFinder: evaluating dynamic queries in a real-estate information exploration system. Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval Copenhagen, Denmark, ACM Press New York, NY, USA.
- Wurman, R. S. (1989). Information Anxiety. Doubleday, New York.

## 10 Appendix – A: Demographics Questionnaire

Age	
Gender	<input type="checkbox"/> Male <input type="checkbox"/> Female
Occupation	
Major/ Area of specialization	
How often do you use computers?	<input type="checkbox"/> Everyday, several hours <input type="checkbox"/> Few times a month <input type="checkbox"/> Few times a week <input type="checkbox"/> Rarely
How long have you been working with software that requires you to use a mouse to point , click, right click, choose menus etc.	<input type="checkbox"/> Few days <input type="checkbox"/> Few weeks <input type="checkbox"/> Few Months <input type="checkbox"/> Several Years  If years, how many years, approx _____
What kinds of software do you use? Please list AS MANY AS you can. (For example word processing like Microsoft Word, image editing software like Adobe Photoshop, email like Yahoo)	

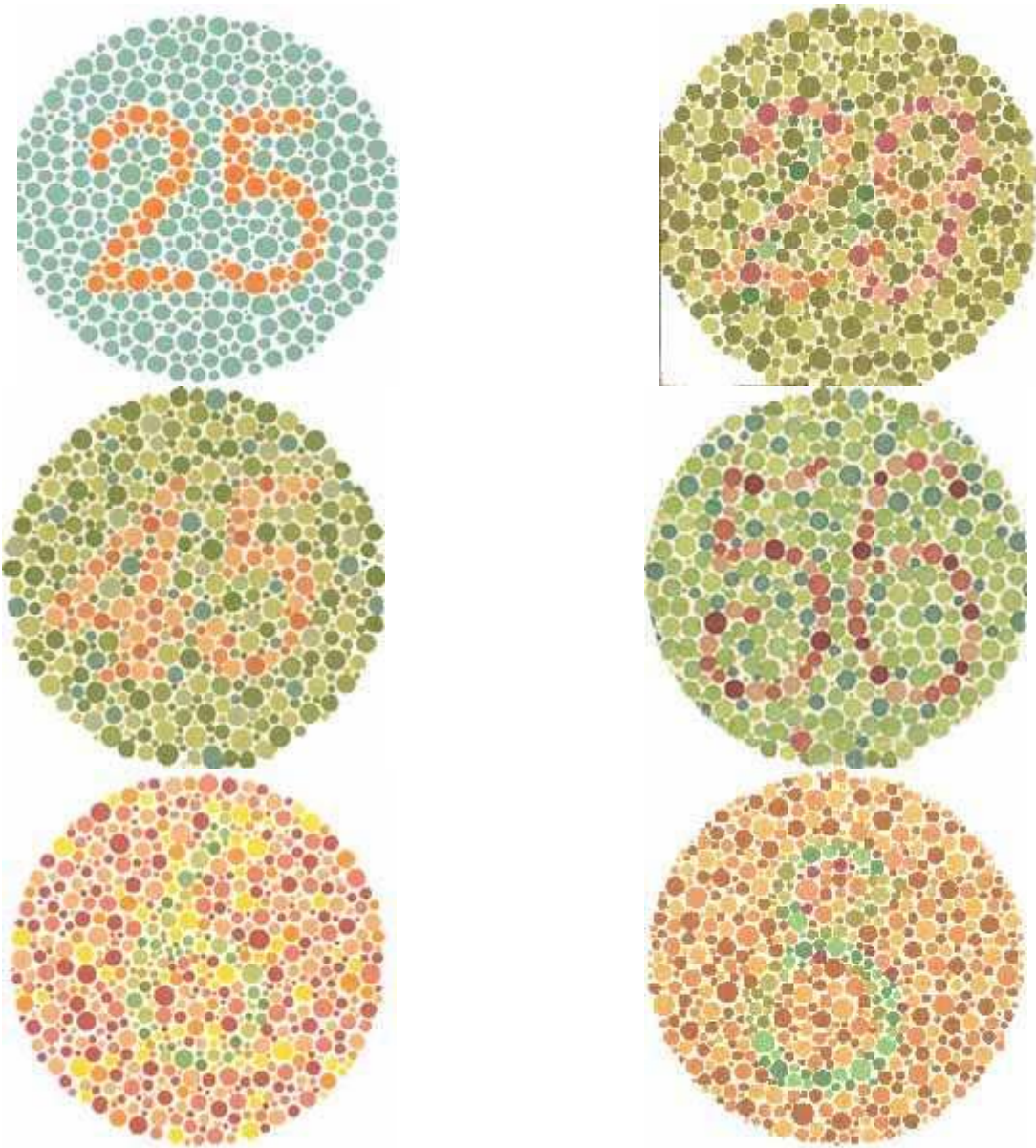


Are you familiar with names of states in USA?	<input type="checkbox"/> None <input type="checkbox"/> Few <input type="checkbox"/> I know several state names <input type="checkbox"/> All
How familiar are you with the map of USA? (We are not evaluating your knowledge. Software should ideally be helpful to you, even if you don't know. Please answer as accurately as possible)	<input type="checkbox"/> Not at all familiar – I can identify 0 states <input type="checkbox"/> Reasonably comfortable – I can identify a few states <input type="checkbox"/> I can identify several states <input type="checkbox"/> I can identify almost all states
Have you ever come across Census data about the USA?	<input type="checkbox"/> No <input type="checkbox"/> Yes
If yes, where have you come across census data about USA? (Check all that apply)	<input type="checkbox"/> No, not at all <input type="checkbox"/> On television <input type="checkbox"/> In newspapers <input type="checkbox"/> On internet <input type="checkbox"/> Other places: _____

<p>How familiar are you with Census data?</p>	<p><input type="checkbox"/> I don't know what is meant by "Census".</p> <p><input type="checkbox"/> I know what Census means, nothing more</p> <p><input type="checkbox"/> I have a vague sense of what kinds of items Census Bureau collects data for.</p> <p><input type="checkbox"/> I have seen Census data about the US states a few to several times</p> <p><input type="checkbox"/> I sometimes use Census data for my knowledge/work</p> <p><input type="checkbox"/> I frequently use Census data for my knowledge/work</p> <p><input type="checkbox"/> Other, explain: _____</p>
---	---

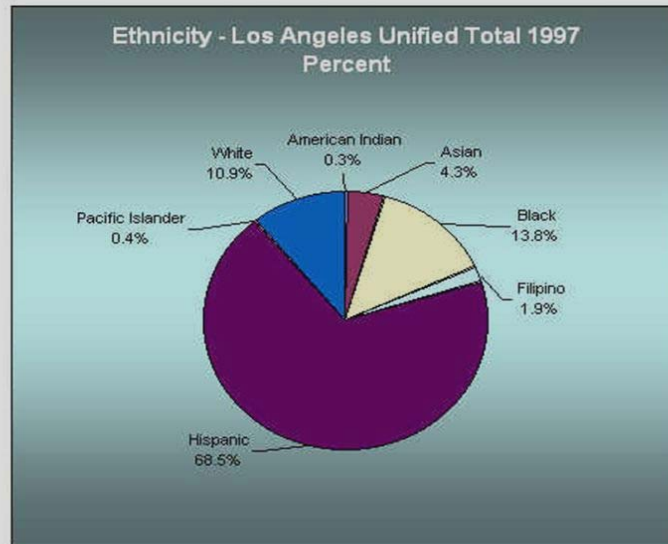
## 11 Appendix B – Screening test

1) Please look at the charts and call out the numbers that you see among the dots.

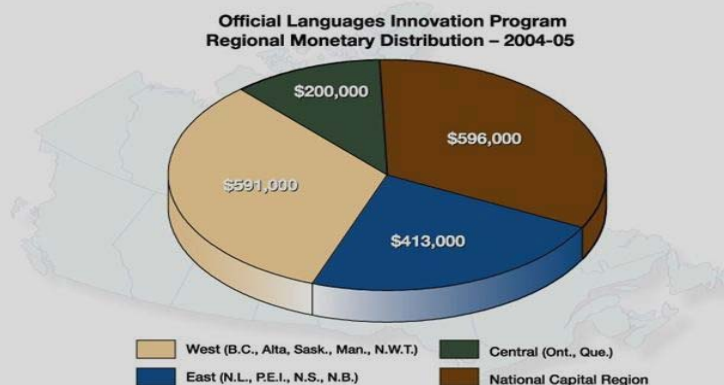


2) Please answer the questions in the following two screens.

Please describe what you understand from this picture



Please use the picture to answer the 2 questions below



- 1) "Central" Region spends how much on the program?
- 2) Which region spends the maximum amount?

## 12 Appendix C: Main Experiment Task List

### PRETASK

This is a visualization software. It helps you obtain census information about states in the USA. Please look at the software and explore and play around. Think aloud as you explore.

### TASK LIST

Now that you are familiar with the software, we will ask you to perform the following tasks. Think aloud as you perform them. If at all you need scratch paper, use the space provided below the questions.

S.N	Task
1	What was the population of California in 1997?
2	Compare the “Personal income [BEA] – per capita [dollars], 1994” values for any 2 states not next to each other. Which had greater per capita income?
3	Of the three states on the west coast, which state had the lowest civilian labor force unemployment rate in 1996?

4	<p>Name the states which had less than 50% white population in 1996. Refer to "Population - percent white, 1996" while you work at this question.</p>
5	<p>Which two states had the highest population percent change from 1990 to 1997? Refer to "Population – percent change, 1990 to 1997" while you work at this question.</p>
6	<p>How many families in Minnesota lived below the poverty line in 1989? Refer to "Families below poverty level in 1989" while you work at this question. This is found under "Poverty" category.</p>

7	<p>Find a state that has</p> <ul style="list-style-type: none"> <li>d) LOW number of “marriages per 1000 population” in 1988 AND LOW number of “divorces per 1000 population” in 1988.</li> <li>e) HIGH number of “marriages per 1000 population 1988” AND HIGH number of “divorces per 1000 population 1988”</li> <li>f) LOW number of “marriages per 1000 population 1988” AND HIGH number of “divorces per 1000 population”.</li> </ul> <p>Note: Marriages and Divorces data are found under “Vital statistics- Births, Deaths, Marriages, Divorces”</p>
8	<p>Which state in the USA had the highest population in 1997?</p>
9	<p>Point to clusters of states on the map having low values for educational attainment. Refer to “educational attainment – percent persons 25 years and over, high school graduates, 1990” while you work at this question.</p>

## 13 Appendix D: Exit Questionnaire

1) You have just used a visualization software. A visualization software is one in which you can view data presented graphically (using maps, bars, pie charts, plot graphs etc.). Have you used any software similar to this one before? Can you think of any software that you have used that was at least partially similar to DataMaps? Please describe:

---

---

---

2) You have used several views at at time (for example, list of attributes and map or map and a table) in DataMaps software. Have you ever worked with applications which require you to use 2 or more views at the same time? Please name them or describe your experience:

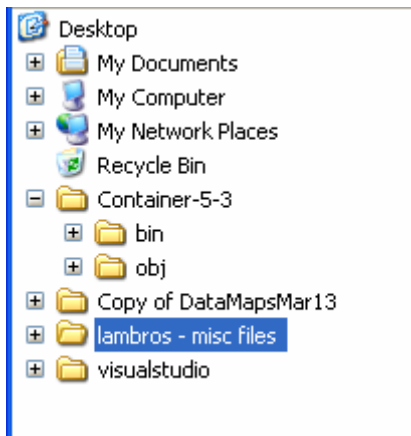
---

---

---

3) Check every item that applies for the following questions. Before performing tasks for this experiment, were you familiar with :

[ ] + signs that expand and - signs that shrink in a tree view as seen in the following figure?



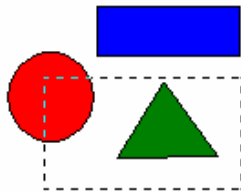


[ ] Right clicking the mouse to see menus on screen? (Example shown below)




[ ] Holding down mouse button and dragging the mouse over regions on the screen?

For example, in the following picture, a region in the picture is being selected by dragging.



[ ] Clicking on table column headers to sort the entire column of numbers?

For example, in the table shown in the following figure, clicking on “Expenses” would sort the rows in descending order.

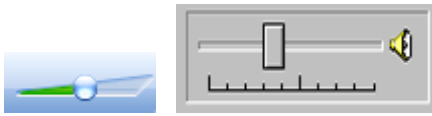
Expenses 
10,000
12,000
12,400
13,030

[ ] Menus on top of software applications? (Example: File->Open, View ->Zoom etc.)

[ ] Interactive maps?

[ ] Interactive tools that help you set value ranges from both sides like in the application you used now.

[ ] Volume control sliders? (Example shown in the figure below)



[ ] Bar graphs?

[ ] Colored maps? (Maps which use colors to mean different things, example: maps in an atlas)

# 14 Appendix E: Informed Consent Form

## **Informed Consent Form** Virginia Polytechnic Institute and State University

Title of Project: Learnability of Datamaps Census Data Visualization

Principal Investigator: Dr. Christopher North

### **I. THE PURPOSE OF THIS RESEARCH/PROJECT**

You are invited to participate in a short experiment on the ease of use of a software tool called “DataMaps.” This study is part of an ongoing research on how to design software for visualization of data to better suits the needs of first-time or novice users.

### **II. PROCEDURES**

You will be asked to perform a few simple data finding tasks using the “DataMaps” software tool, using the mouse. You will be asked to think aloud, as you perform the tasks. We may capture the screen shots of your mouse movements on screen and also record your face and voice as you speak, on video. You will also be asked to fill out a questionnaire relating to your background with such systems, and to take a short test of graph-interpretation ability. You may be asked questions during and after the evaluation, in order to clarify our understanding of your evaluation.

The session is expected to last around half an hour to forty five minutes. The tasks are not very tiring, but you are welcome to take rest breaks if needed. You may also terminate your participation at any time, for any reason.

You will be given full instructions before every task. It is important that you understand the instructions before beginning each task. If anything is unclear, be sure to ask questions.

### **III. RISKS**

The proposed experiments are straightforward tests of performance using standard desktop software and mouse. Participation involves sitting in front of a desk top and using the mouse to perform tasks while thinking aloud. The tasks are not stressful and you may take breaks at any time if you wish. Your face and voice may be recorded on video.

### **IV. BENEFITS OF THIS PROJECT**

Your participation in this project will provide information that may be used to improve the design of visualization software. No guarantee of benefits has been made to encourage you to participate. You may receive a synopsis summarizing this research when completed. Please leave a self-addressed envelope with the experimenter if you would like to receive a copy of the results. You are requested to refrain from discussing the evaluation with other people who might be in the candidate pool from which other participants might be drawn.

### **V. EXTENT OF ANONYMITY AND CONFIDENTIALITY**

The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a subject number will identify you during analyses and any written reports of the research.

The experiment may be recorded. If it is recorded, the recordings will be stored securely, viewed only by the experimenters (Christopher North and Sujatha Krishnamoorthy), and erased after 3 months. If the

experimenters wish to use a portion of the recording for any other purpose, they will get your written permission before using it. Your signature on this form does not give them permission to show your recording to anyone else.

## **VI. COMPENSATION**

Your participation is voluntary and unpaid.

## **VII. FREEDOM TO WITHDRAW**

You are free to withdraw from this study at any time for any reason.

## **VIII. APPROVAL OF RESEARCH**

This research has been approved, as required, by the Institutional Review Board for projects involving human subjects at Virginia Polytechnic Institute and State University, and by the Department of Computer Science.

## **IX. SUBJECT'S RESPONSIBILITIES AND PERMISSION**

I voluntarily agree to participate in this study, and I know of no reason I cannot participate. I have read and understood the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project

---

Signature

---

Date

---

Name (please print)

---

Contact: phone or address or

---

email address (OPTIONAL)

Should you have any questions about this research or its conduct, you may contact:

Investigators: Christopher North 540-231-2756  
Assistant Professor, Computer Science Department  
email: [north@cs.vt.edu](mailto:north@cs.vt.edu)

Sujatha Krishnamoorthy  
Student, Computer Science Department  
Email: [sukrish2@vt.edu](mailto:sukrish2@vt.edu)