Project Number: MQP-MOW-1101

IMPROVED VISUALIZATION OF NETWORKED STORAGE METRICS

A Major Qualifying Project Report

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

____
Xiao Du

____
Zhaochen Liu

____
David Rollé

Date: October 21, 2011

Approved by:

____

Professor Matthew O. Ward, Advisor

____

Professor Emmanuel Agu, Advisor

Abstract

This project's goal was to create visualizations for EMC which resolve problems with complexity in the field of networked storage management. These visualizations were aimed at representations of the network's storage structure, visualizations the data through network, usage of space over time, and usage by a group/location. Each design was made distinct from the other designs, to demonstrate different methods of visualization. The team evaluated these designs with EMC employees, and was received well.

Authorship Page

Abstract	DR
Acknowledgement	ALL
Executive Summary	DR
1 Introduction	DR
1.1 Use Cases	ZL
1.2 Difficulties	ZL
1.3 The Goal of the MQP	DR
2 Related Work	XD, ZL
2.1 Space-Filling Methods: Treemap	XD, ZL
2.2 Space-Filling Methods: Sunburst	XD, ZL
2.3 Cone Tree	ZL
2.4 Area Graph and Stacked Area Graph	ZL
3 Background	ZL
3.1 Background Technology	ZL
3.2 Current Solutions	ZL
4 Design	DR
4.1 Design Introduction	DR
4.2 Previous Designs and Their Faults	DR
4.3 Preliminary Design Concepts	DR
4.4 Enhancing the Prototype Design	DR
5 Implementation	XD
5.1 Chargeback & Array Visualization	DR
5.2 Capacity usage Breakdown Visualization	ZL
5.3 Consumption Overview	XD
6 Evaluations	ZL
6.1 Study Subjects	ZL
6.2 Feedback from Subjects	ZL
6.2.1 Subject 1	ZL
6.2.2 Subject 2	XD
6.2.3 Subject 3	DR
6.2.4 Subject 4	ZL
6.2.5 Subject 5	XD
6.3 Summary of Evaluation	DR
7 Conclusion	ZL
7.1 Major Accomplishment	ZL
7.2 Evaluation Summary	ZL
7.3 Future Work	XD

Appendix A	ALL
Appendix B	ALL
Appendix C	ALL
Editing Report	ALL
Implementation of the Prototype	
Chargeback Visualization	DR
Capacity Usage Breakdown Visualization	ZL
Consumption Overview	XD

Executive Summary

This paper dealt with the difficulties related to the design and development of Storage Area Network (SAN) materials that are used to create multiple backups of data. The team began by analyzing data from EMC's past experiences with SAN visualizations.

EMC is in the business of data "protection," which is the replication and remote copying of data in an effort to keep problems, such as hard drive failures, power outages, or even large-scale catastrophic failures, such as the destruction of a datacenter, from affecting a business's ability to continue operating. However, a complex network system such as this is immensely complicated, often with multiple layers of redundancy and abstractions. EMC's current designs include tables and a basic, low-information topology map.

The goal of this MQP was to take several use cases presented to the MQP team, and create simple, easy-to-understand visualizations that would allow customers which use EMC's products to follow the complexity of the system represented. Eventually, the team settled on three use cases allowing a wide breadth of designs to be considered in a short amount of time. These use cases were chargeback visualization, with another section on array visualization, usage by groups/locations, and usage over time. Each member of the MQP team took a different use case and tackled them independently.

The first use case was a visualization, the purpose of which was to better represent how much space a host is using, including local and remote replicas, also known as a “chargeback” visualization. The purpose of this use case was to aid the consumers of StorageScope in understanding how the data traversed the storage network. This traversal included the logical, abstract forms the data takes, described as Logical Unit Numbers (or LUNs), which are represented as internal SCSI disks to the host computer, and also physical forms, as disks grouping into RAIDs, then grouped into arrays. StorageScope is EMC’s proprietary technology for which these designs were being developed.

This design was ultimately implemented in an HTML5 <canvas> element, using many of the features common in HTML5 today. The goal was to render it in a three-dimensional format, but restrictions in the technology forced the implementation to be designed in a 2D environment with 3D-illusion effects. The array visualization was also constructed in a manner similar to the chargeback, as they were similar in purpose and functional design. The final design took on the look of a cylinder, with the sides representing the various states data could exist in. Difficulties in the development included the problems with scalable designs, and the simplification of broad specifications that technically fall under, down to a basic, easily understood representation.

The second use case was usage over time. This was used to visualize the user’s storage system with time being a consistent factor in the changing of a storage volume's data. Since the data that users consume is consistently growing the more that a user has access to the system, the storage device needs to be able to hold all of the company's data, while never running out of storage. In this case, a method to track the entire SAN, or a portion of it, depending on what the user would like to monitor, would assist in the prediction of future issues.

This was rendered using two APIs; one contained a circular visualization used to render percentage of the dataset, represented in used, free, and allocated space, while the other visualization was a trend graph that rendered the data directly. The first API rendered the design in a treering-like design, with each ring further out representing a later point in time, just as a tree gains rings with each year. The rings were then divided into the three measured values, to represent the percentages. The trend graph was then linked to the ring graph such that, when a section of the trend graph was highlighted, the corresponding ring became lit, and vice versa. This provided a comprehensive, easily-interpreted and interactive view which the user could manipulate without difficulty.

The third use case was usage by group/location. This use case was directly responsible for visualizing every portion of the SAN that is used by a group, location, application, service level, or additional members. The purpose of this use case is to understand how different values are affecting the overall integrity of the storage network. If a group of users are dumping large amounts of data onto the network rapidly, while being allocated only a small amount of space, then the storage administrator needs to be able to respond to this. On the other hand, if a user has a large section of data allocated to them, but only a small section is used at any given point in time, the storage administrator might be able to minimize the total size available.

Measuring these values was accomplished by a treemap, a design able to represent the data of two or more sets, without complicating the data schema. The user was also allowed to pick and choose what pieces they were most interested in for more information. The reasoning behind a treemap was that a treemap had the necessary complexity, spatial orientation, and visual appeal the users required to understand the data without difficulty, while maintaining the dataset’s individual distinctiveness that was necessary for a user to differentiate between separate chunks of data.

The prototypes were developed based on the design, and the design was put before a select group of five EMC employees, a sixth being the sponsor, who were familiar with the current setups that EMC has. The results were favorable, and while each visualization had noted flaws, they were, on the whole, considered to be a successful representation of the data presented.

Acknowledgement

The MQP Team would like to acknowledge the following people who, without them, this project would not have been possible.

Professor Matthew O. Ward, WPI

Professor Emmanuel Agu, WPI

Dan Norton, EMC Liaison to MQP Team

Alyce Buchenan, who pushed David to do as much as he could.

Table of Contents

Abstract 2

Authorship Page 3

Executive Summary 5

Acknowledgement 9

Table of Contents 10

1 Introduction 14

1.1 Use Cases 15

1.1.1 Use case 1 15

1.1.2 Use case 2 15

1.1.3 Use case 3 16

1.1.4 Other Uses Cases 16

1.2 Difficulties 17

1.2.1 Large quantity of data 17

1.2.2 Specific problems 18

1.2.3 Different details required 18

1.3 The Goal of the MQP 18

2 Related Work 20

2.1 Space-Filling Methods: Treemap 20

2.2 Space-Filling Methods: Sunburst 22

2.3 Cone Tree 26

2.4 Area Graph and Stacked Area Graph 28

3 Background 30

3.1 Background Technology 31

3.2 Current Solutions 33

4 Design 35

4.1 Design Introduction 35

4.1.1 Use Case 1 36

4.1.2 Use Case 2 36

4.1.3 Use Case 3 37

4.1.4 EMC Technology and Use Cases 38

4.1.4.1 Use Case 1 Technology Breakdown 38

4.1.4.2 Use Case 2 Technology Breakdown 39

4.1.4.3 Use Case 3 Technology Breakdown 40

4.2 Previous Designs and Their Faults 41

4.2.1 Difficulties with Representations 41

4.2.2 Bar Graph Representations 42

4.2.3 Pie Charts 43

4.2.4 Tables 44

4.2.5 Maps 46

4.3 Preliminary Design Concepts 47

4.3.1 First Use Case Designs 48

4.3.2 Second Use Case Design 51

4.3.3 Third Use Case Design 53

4.4 Enhancing the Prototype Design 55

4.4.1 User Interface Design 55

4.4.2 Topology Map Design Enhancements 56

4.4.3 Treemap Design Enhancements 60

4.4.4 Usage over Time Design Enhancements 62

5 Implementation 65

5.1 Chargeback & Array Visualization 65

5.1.1 Technology – HTML 5 and Javascript Code 66

5.1.2 UI Design Implementation 69

5.1.2.1 Mouse Controls 69

5.1.2.2 Breadcrumbs 71

5.1.2.3 Rapid Navigation 72

5.1.2.4 Scalability 72

5.1.2.5 Background Sectioning 72

5.1.3 Chargeback Implementation 73
5.1.4 Array Implementation 75
5.2 Capacity Usage Breakdown Visualization 75
5.2.1 Some Major Technologies 76
5.2.1.1 DrasticTreemap 76
5.2.1.2 Google Chart API 78
5.2.1.2.1 Motion Chart 79
5.2.1.2.2 Bar Graph & Sparkline 80
5.2.1.3 JavaScript & jQuery Tools 81
5.3 Consumption Overview 83
5.3.1 Adobe Flex 83
5.3.2 Axiis 84
5.3.3 Degrafa 85
6 Introduction to Evaluations 88
6.1 Study Subjects 88
6.1.1 Study Protocol 88
6.2 Feedback from Subjects 89
6.2.1 Subject 1 89
6.2.2 Subject 2 91
6.2.3 Subject 3 93
6.2.4 Subject 4 95
6.2.5 Subject 5 97
6.3 Summary of Evaluation 99
7 Conclusion 101
7.1 Major Accomplishment 101
7.2 Evaluation Summary 103
7.3 Future Work 104
7.3.1 Modifications to Design 104
7.3.2 More Use Cases 105
7.3.3 User Study 105

7.3.4 Use Real Data ... 106

Bibliography ... 107

Appendix A - Sample User Study Consent Form ... 112

Appendix B - User Study Protocol ... 115

Appendix C - Sample User Study Questionnaire & Feedback ... 117

1 Introduction

Networked storage is a complex and difficult task to understand, much less represent visually with ease of understanding and accuracy. However, since replication and backup of gigabytes, terabytes, or even petabytes of data, either in a local array or over distances of kilometers or more, preserves data from loss or destruction, monitoring the system's operations at all times is imperative. The most important requirements of any system that observes such a complex network, known as a Storage Area Network (SAN), should be simplistic in design, while readily detailing all the required information. The quantity of data that is monitored, however, would require multiple different designs.

The difficulty of representing data in an efficient manner, while containing the myriad of variables present in a system as complex as a SAN, representing the interdependency of these variables, and presenting it in a visually appearing manner, has challenged EMC's technicians for many years. The focus of this project was developing visualizations of use cases EMC presented, starting from design versions and then creating a functioning prototype with interactive interface. A large problem with the creation of these designs is that the user often has to work very hard to understand what they are looking for, and if the user has difficulty piecing together the complexity of a networked storage system, then the visualizations need to alleviate this problem.

1.1 Use Cases

The current problem that the Ionix Storage Management group encountered was finding a good way to visualize the supply chain in a way that was easy to understand, while also giving the user the information that they needed. When EMC's customers purchase data storage systems, this visualization software will also be deployed, in order to help them setup and monitor the storage system.

Therefore, the main objective of this project is to design and develop visualization methods that can give EMC's engineers some possible data visualization ideas. Feedback will be collected from user studies at later stage of the project.

Some of the use cases that EMC provided focused on the development of the visualizations based on three questions: who were the primary users, what the essential or desired features were, and what kind of data was involved in the designs.

1.1.1 Use case 1

The storage administrator should have a good overview of the whole system, and be able to explain to his/her boss how the large amount of raw storage that was purchased is different from the total usable storage. The admin should also be able to explain why this disparity occurs, such as due to fragmentation or un-configured space.

1.1.2 Use case 2

Users need to have a decent knowledge of how the capacity is used. For instance, a specific amount of storage space can be used by the engineering group, while another sliver of storage can be used by the marketing group. These designs must be able to show the users which groups, separated into metrics, have which space. One of these metrics, for example, is the location of storage arrays.

1.1.3 Use case 3

A third critical component to storage administration is knowing the consumption speed of a storage system's resources over time, because the admin needs to be able to anticipate and prevent outages in the SAN caused by running out of space. This way, a storage administrator will be able to predict when it is necessary to purchase more storage.

1.1.4 Other Uses Cases

There are additional uses cases provided by the sponsor. But the MQP team chose to deal with these three use cases because they are the most essential problems, as suggested by Mr. Norton. After solving these use cases, other uses cases can be dealt using a similar approach.

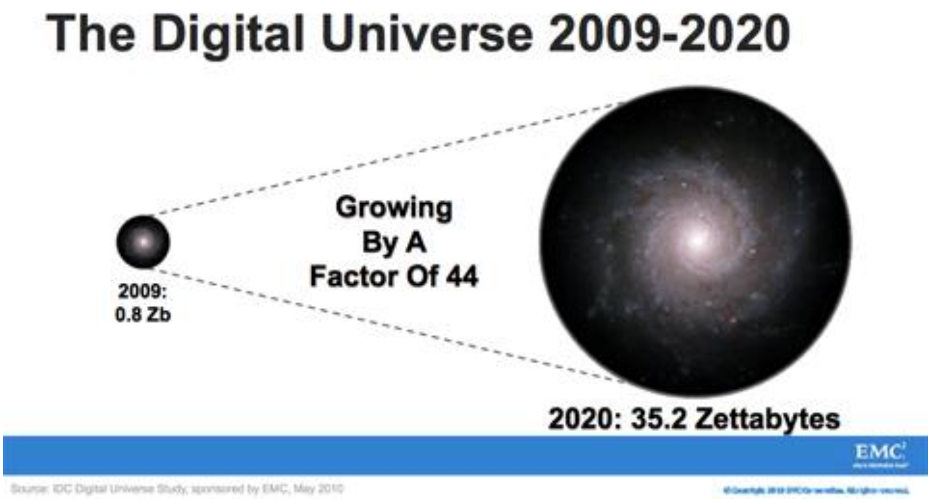
One of the other use cases is that the storage administrator wants to have the ability to change the slope of the trend line. This way, he can easily tell how the consumption changes if some events happen. Another use case is that the storage administrator wants to know which are the systems that appear to be in use but aren't really. Some reasons can be that server hasn't recognized the LUN or there is no logical zone. The last use case is how to clearly show customers the services level they are purchasing.

1.2 Difficulties

Some difficulties and obstacles can be identified before starting to solve the problem.

1.2.1 Large quantity of data

EMC's customers deal with colossal amounts of data. According to EMC's official website and SearchStorage.com[1], as of 9/28/2011, some of EMC's biggest customers include companies such as HP, Microsoft and Bank of America. These industry-leading companies require large amounts of information to be held in secure, reliable, and redundant setups. Also, EMC predicts the growth of data will be very rapid, as shown in Figure 1-1. Therefore, scalability is one of the critical factors for consideration while designing the system. At a minimum, it should have some features, such as filters, to address the problem.



Source: IDC Digital Universe Study, sponsored by EMC, May 2010

Figure 1-1 Image Depicting Predicted Data Growth From 2009-2020 [2]

1.2.2 Specific problems

According to the use cases discussed earlier, the actual problems that were under consideration were specific to EMC's products, most notably the supply chain. Due to confidentiality, EMC and other storage companies generally do not release information regarding these designs to the public. Based on these two factors, online resources are usually very limited in their discussions of the topic. Consequently, many of the solutions that were developed had to be developed from a unique perspective, without additional support from previous designs.

1.2.3 Different details required

Different people need to interact with the software at different levels. For example, an upper management team may not care about detailed information of a specific LUN. In fact, the leader might just need a high level overview of the entire SAN. On the other hand, storage administrators, who are meant to be in charge of the entire system and delegate space to different groups in a company, would need specific information of each individual device. Therefore, the potential system should need the ability to adjust the perspective displayed to the user at any given point.

1.3 The Goal of the MQP

The purpose of this MQP was to visually present the use cases which were provided, in such a way that the potential user groups of a SAN network would be able to explore and interact with the data, while still allowing the user group to readily achieve their intended results with ease and speed. This would require a focus on speed and simplicity in design, as well as scalability. Multiple visualizations must be created to maximize the variety of designs which EMC could draw from in the future.

2 Related Work

There is not much work that has been done in relation to this specific topic. However, there are a lot of resources about visualizing multivariate data that aid in the designing of new visualizations for this project. This section discusses a few of these relevant ideas.

2.1 Space-Filling Methods: Treemap

A treemap is a visualization method for displaying tree-structured data using nested rectangles [3] as shown in Figure 2-1. A treemap is also a space-filling approach to the visualization of hierarchical information structures. Space-filling techniques make maximal use of the display space [4]. This is accomplished by using juxtapositioning to imply relations. In the book *Interactive Data Visualization: Foundations, Techniques and Applications*[4], Ward, Grinstein and Keim, the authors, provided the pseudo code for generating a treemap is provided, as shown in Figure 2-2.

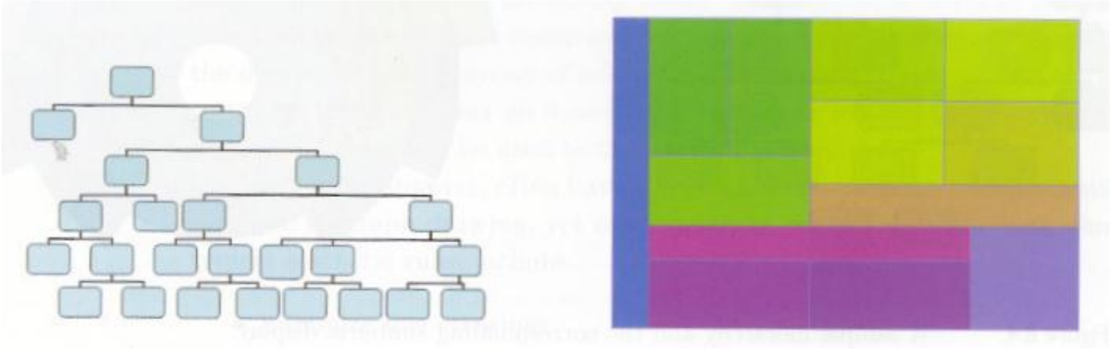
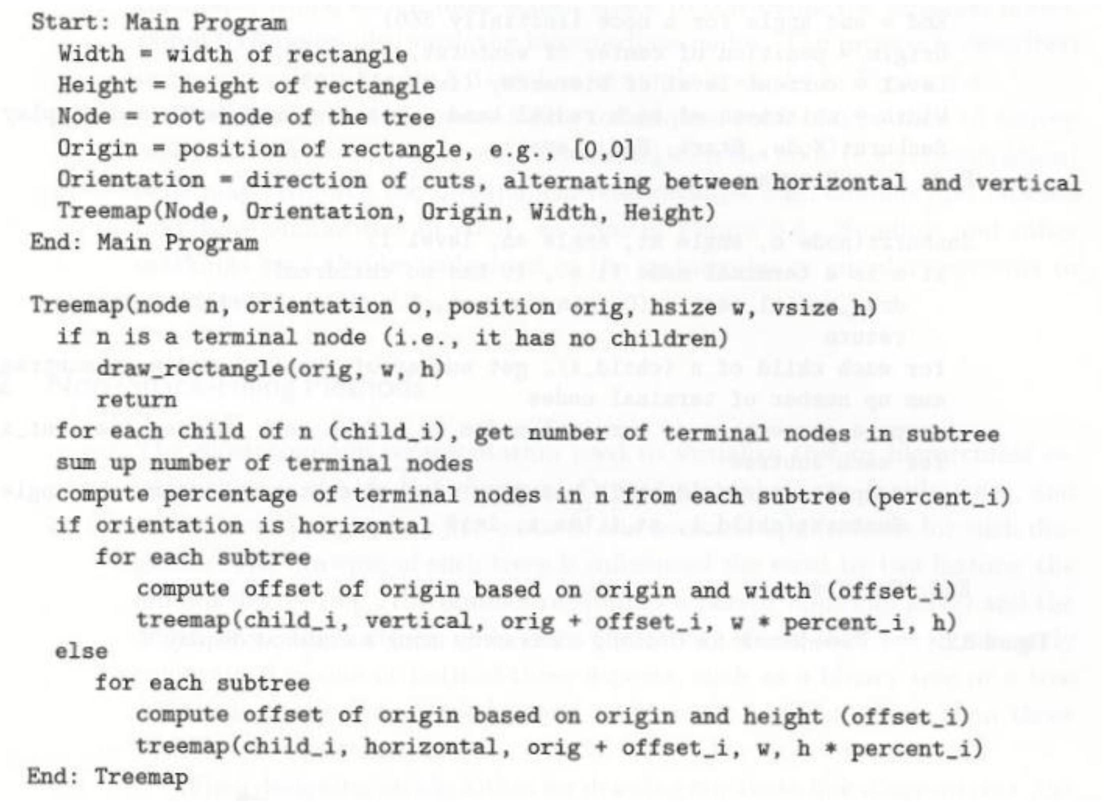


Figure 2-1 A sample hierarchy and the corresponding treemap. [4]



```
Start: Main Program
  Width = width of rectangle
  Height = height of rectangle
  Node = root node of the tree
  Origin = position of rectangle, e.g., [0,0]
  Orientation = direction of cuts, alternating between horizontal and vertical
  Treemap(Node, Orientation, Origin, Width, Height)
End: Main Program

Treemap(node n, orientation o, position orig, hsize w, vsize h)
  if n is a terminal node (i.e., it has no children)
     draw_rectangle(orig, w, h)
     return
  for each child of n (child_i), get number of terminal nodes in subtree
  sum up number of terminal nodes
  compute percentage of terminal nodes in n from each subtree (percent_i)
  if orientation is horizontal
     for each subtree
        compute offset of origin based on origin and width (offset_i)
        treemap(child_i, vertical, orig + offset_i, w * percent_i, h)
  else
     for each subtree
        compute offset of origin based on origin and height (offset_i)
        treemap(child_i, horizontal, orig + offset_i, w, h * percent_i)
End: Treemap
```

Figure 2-2 Pseudocode for drawing a hierarchy using treemap. [4]

In the paper Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information Structures[4], the authors, Johnson and Shneiderman, showed how the treemap uses 100% of the available display space to present the full hierarchy onto a rectangular region. Humans have the ability to recognize spatial information display in a picture and notice the relationship between elements quickly. This highly developed visual ability allows people to grasp the content of a picture much faster than reading text [4].

In addition, Johnson and Shneiderman compared the Venn diagram representation and Treemaps. In the Venn diagram Figure 2-3, there is a large amount of space between each small circle that is wasted. On the other hand, Figure 2-4 showed that the rectangles fit each other better, resulting in a Treemap.

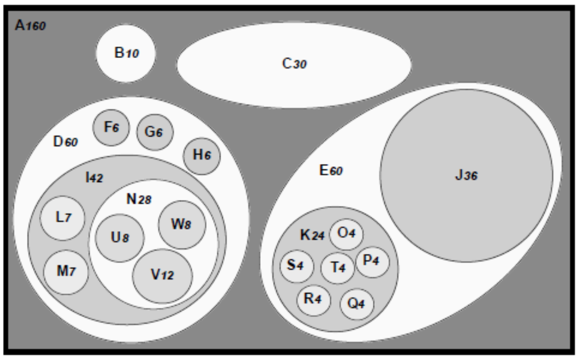


Figure 2-3 Circular Venn Diagram[3]

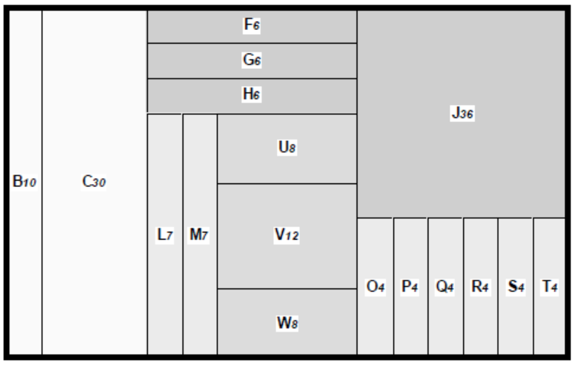


Figure 2-4 Rectangular Venn Diagram[3]

Treemap is relevant to this MQP because it is a good way to represent hierarchical data. It can be used to display capacity usage breakdown. More details will be covered in Section 4.

2.2 Space-Filling Methods: Sunburst

Ward, Grinstein and Keim, indicated another way to convey hierarchy: to divide space radially [4]. Radial space-filling hierarchy visualizations, sometimes referred to as sunburst displays [5], have the root of the hierarchy in the center of the display and use nested rings to convey the layers of the hierarchy. Each ring is divided based on the number of nodes at that level. As Ward, Grinstein and Keim pointed out, the difference between treemaps and radial techniques is that unlike treemaps, which assign most screen space to conveying the terminal nodes, radial techniques also show the intermediate nodes. An example of sunburst display is shown in Figure 2-5.

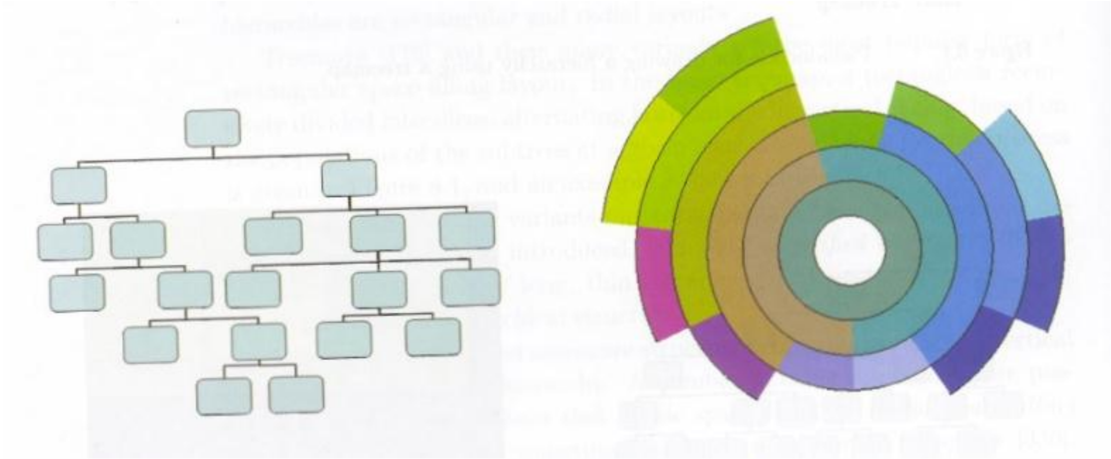


Figure 2-5 A sample hierarchy and the corresponding sunburst display. [4]

Similarly, Stasko and Zhang described radial space-filling visualizations meant to depict information hierarchies. Several different node edge tree layout algorithms were developed ranging from simple 2D planar algorithms [6] to 3D algorithms [7] to hyperbolic methods [8]. The authors developed three visualization techniques that provided flexible browsing of the display. With these techniques, the viewers were able to examine smaller datasets in greater detail while providing context within the frame of the information hierarchy.

The design guidelines were to maintain a balance between the visibilities of both the hierarchy overview and the detailed focus display. At the same time, avoid the use of multiple windows or scrollbars, if possible that would force viewers to mentally connect separated areas or lose context of the entire space, and allow the viewer to easily track display changes between a global view and a detailed view of the peripheral nodes.

Based on these guidelines, the researchers provided three methods for visualizing the data: angular detail method in Figure 2-6, detail outside method in Figure 2-7, and detail inside method in Figure 2-8. After analyzing the strengths and weakness of each method, the Angular Detail method was considered to be more flexible than the other two methods.

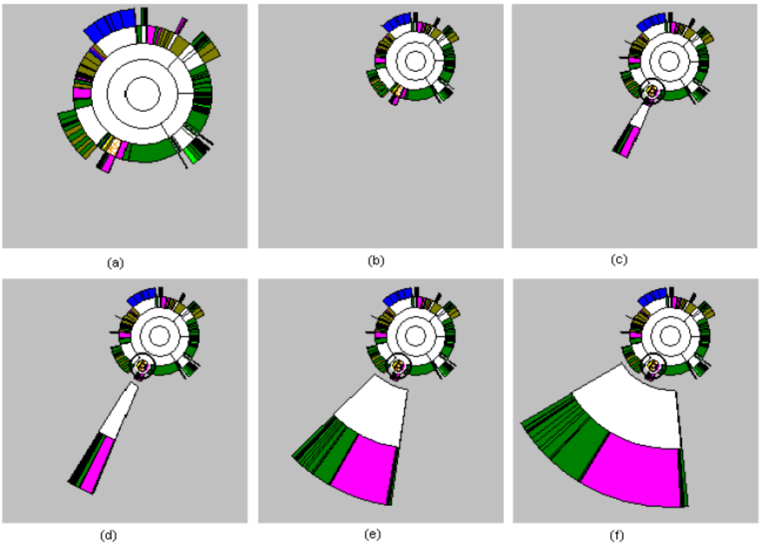


Figure 2-6 Sequence of Snapshots from the Angular Detail Method. The overview is shrunk and pushed to the opposite side from where the focus region grows outward.[5]

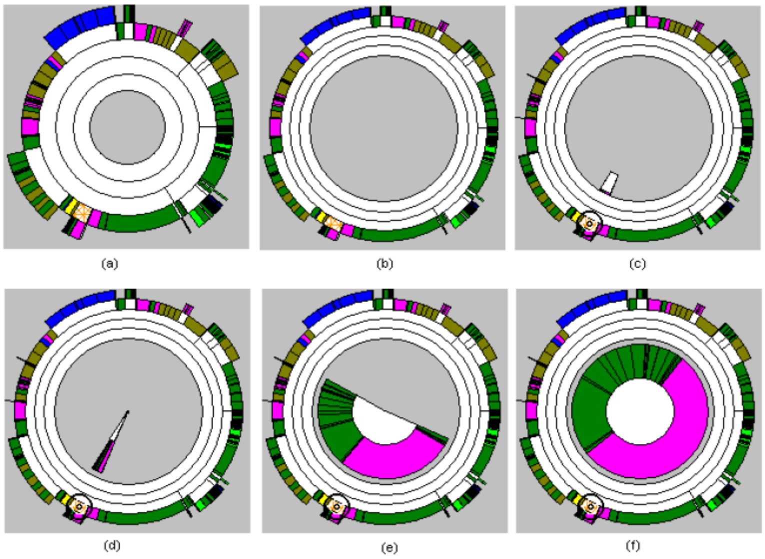


Figure 2-7 Sequence of Snapshots from the Detail Outside Method. The overview shrinks into the center and the focus area is drawn in a ring around the overview.[5]

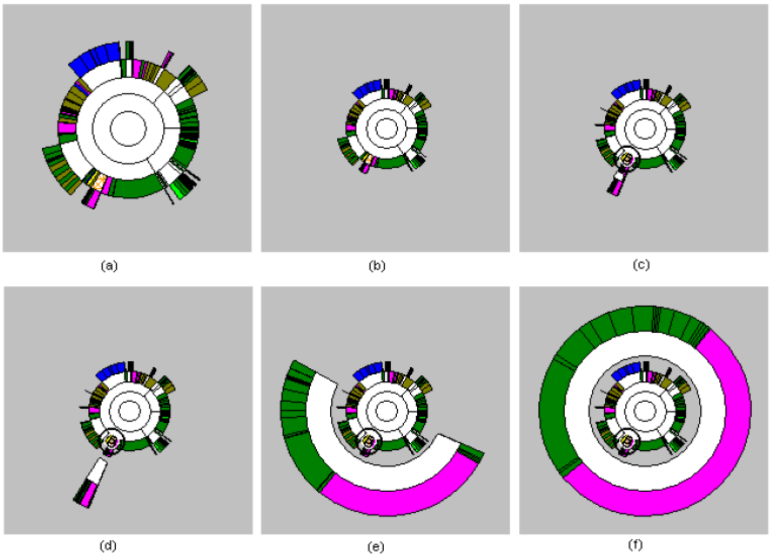


Figure 2-8 Sequence of Snapshots from the Detail Inside Method. The overview is shrunk and pushed outward and the focus region is drawn in the center circular area.[5]

These ideas are relevant to this MQP because Stasko and Zhang provided a way to depict information hierarchies that would be helpful to the MQP team to analyze and develop the visualization of the Networked Storage Metrics.

2.3 Cone Tree

In the book, *Interactive Data Visualization: Foundations, Techniques and Applications,* the authors suggested that the most common representation used to visualize tree or hierarchical relationships is a node-link diagram, such as organizational charts and family trees.

However, when the data became massive, the trees will become very complex and hard to read. Ward, Grinstein and Keim suggested using a third dimension for large trees, supplemented with tools for rotation, translation, and zooming. One of the most well-known of such techniques is called a cone tree [9]. In a cone tree layout, the children of a node are arranged radially at evenly space angles and then offset perpendicular to the plane. The authors suggested that the density of the display and the level of occlusion are two factors that are worth considering. They argued that the radius should be inversely proportional to the depth of a node in the tree. Figure 2-9 is an example of a cone tree.

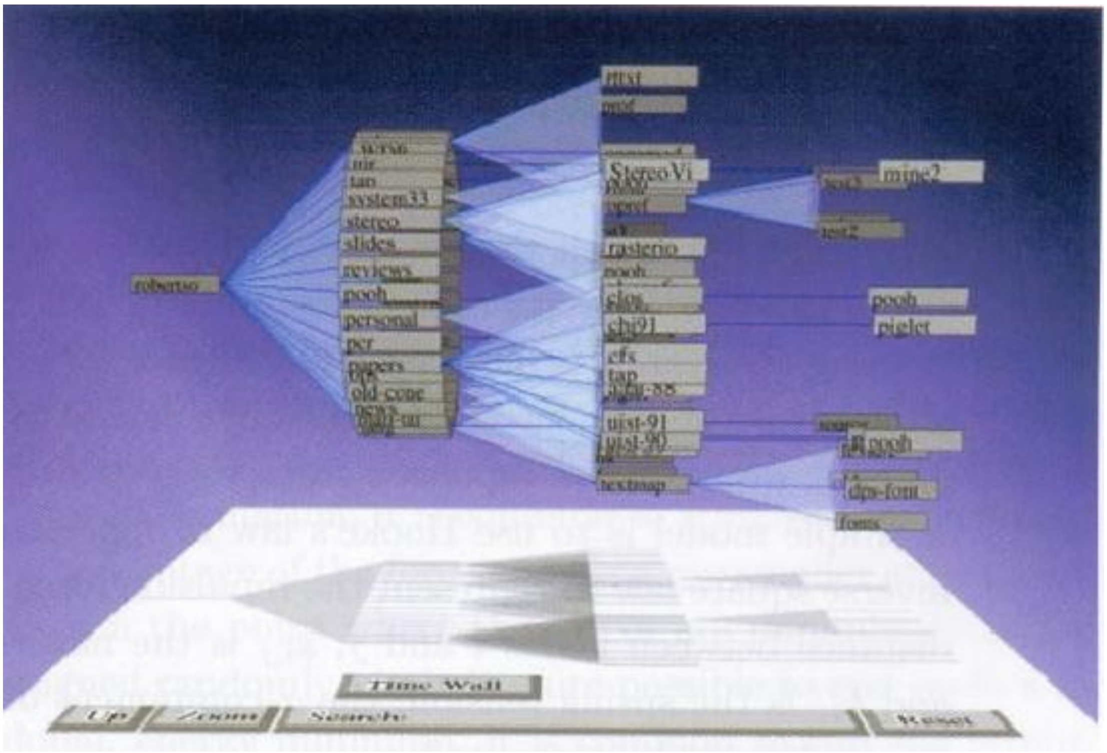


Figure 2-9 An example of a hierarchy displayed with a cone tree. [9]

Alexandru Telea[10] , the author of *Data Visualization – Principles and Practice*, said that the advantage of the cone layout is that its 2D rendering maybe more compact than other layouts, when viewed from an optimal angle. Also, he thought the visual separation of different subtrees may be easier when interactively rotating the viewpoint around the cone layout. Figure 2-10 is an example that the author provided.

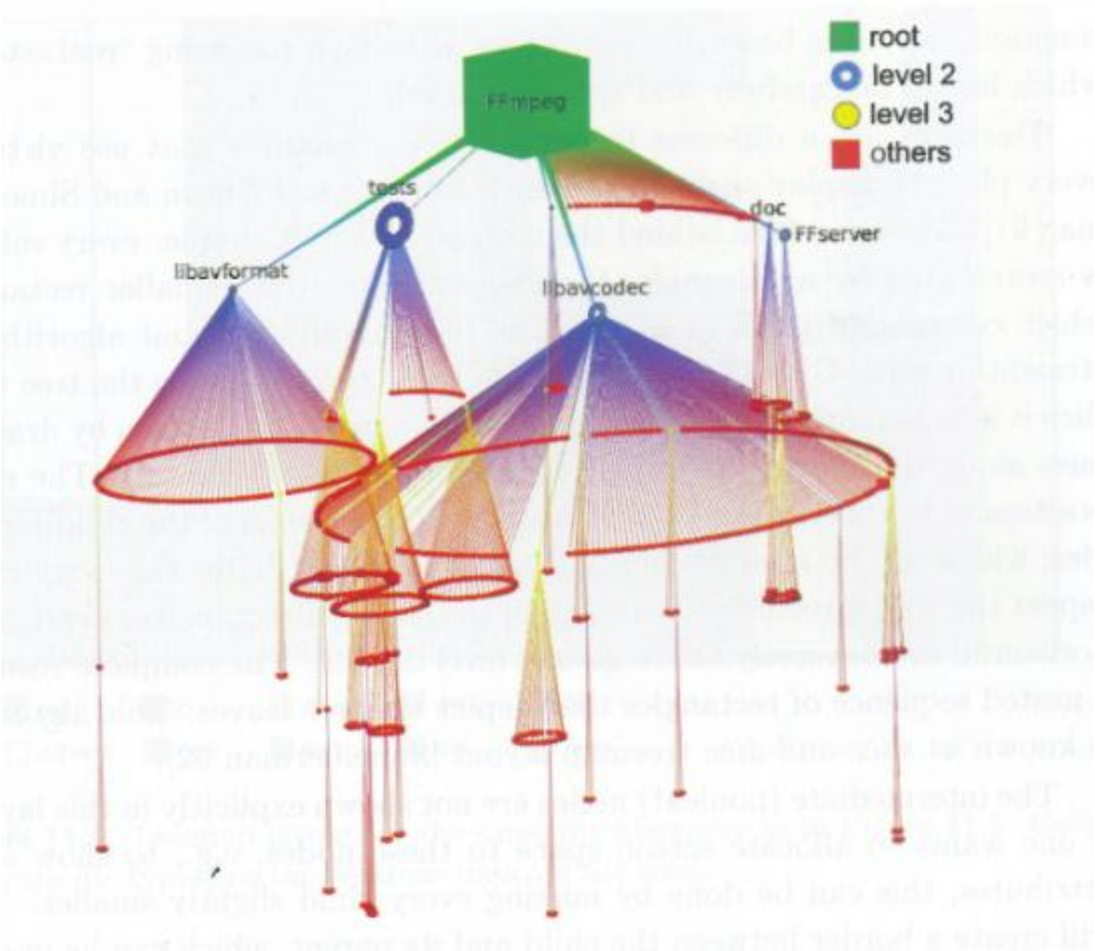


Figure 2-10 Cone-tree layout for a file hierarchy. [10]

The cone tree design is relevant to this MQP because it provides one solution to deal with large and complicated data. EMC's storage systems may include thousands of elements. The MQP team can use such ideas to deal with the use cases.

2.4 Area Graph and Stacked Area Graph

Robert Harris [11] , the author of *Information Graphics – A comprehensive Illustrated Reference*, suggested that an area graph is similar to basic line graph. The difference is that the area between the data curve and the horizontal axis is filled in an area graph. He pointed out that area graphs are generally not used to convey specific values. Instead, they are most often used to show trends and relationships.

A variation of an area graph is a stacked area graph. In a stacked area graph, multiple data series position on top of one another. An important variation of the stacked area graph is the 100% stacked area graph that displays percent-of-the-whole data. A stacked area graph is better at showing the weight of each element. It cannot show the actual values. The author illustrated the difference between these three graphs using Figure 2-11.

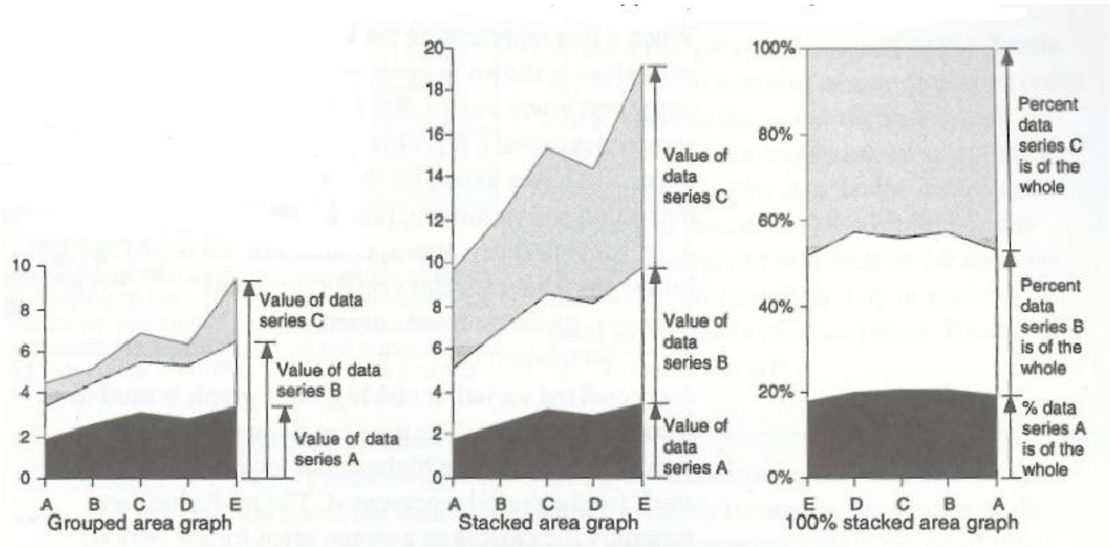


Figure 2-11 Comparison of grouped, stacked, and 100% stacked area graphs. [11]

There are a few uses cases in this MQP where percentage of each element is important. The team may apply some ideas and techniques introduced in Harris' book.

3 Background

This project was sponsored by EMC Corporation, a leading company that develops, delivers, and supports information infrastructure and virtual infrastructure hardware, software, and services.[12]

The MQP team worked with the Ionix Storage Management (ISM) group in EMC in order to develop new methods of depicting the data contained in EMC's storage systems. One main responsibility that Ionix Storage Management had was to collect, correlate, and graphically present performance information for a user's storage infrastructure.[13] ISM used integrated capacity utilization reporting and trend analysis across heterogeneous storage infrastructures. Their goal was to help the users configure, discover, and manage performance across complex physical storage and cloud environments without encumbering them with unintentionally complex data.

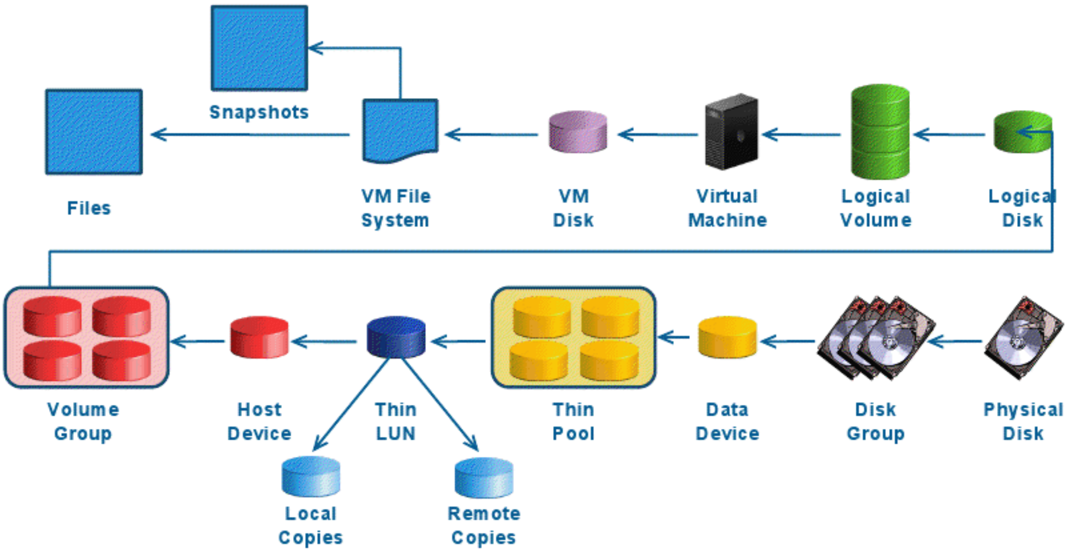


Figure 3-1 EMC Storage Supply Chain Depiction [14]

Figure 3-1 illustrates how the supply chain moves from the physical disk that the data is stored on, to the computer that the user works with. The supply chain also explains the architecture of EMC core technology. In order to gain a better understanding of the supply chain, EMC has assigned terms to the various structures throughout the network.

3.1 Background Technology

Storage Area Network (SAN)[15]**:** A network whose primary purpose is the transfer of data between computer systems and storage elements. EMC uses SAN storage systems consisting of storage elements, storage devices, computer systems, and/or appliances, plus control software which communicates with arrays and hosts over a network.

Logical Unit Number (LUN)[16]**:** A unique number relative to an array host director port or storage group. The LUN is used to differentiate between peripheral devices (each of which is a Logical Unit) that are accessed through a particular array host director port. A LUN is represented as a standard internal disk to the host computer.

Thin Pool [17]**:** A collection of storage capacity that is linked to host visible logical devices, and is dynamically allocated as needed. This is often pieced together from the total storage space of a disk array.

Disk Array [18]**:** A disk storage system that contains multiple disk drives, usually in the form of RAID groups. Figure 3-2 shows a series of EMC Clariion CX500 Disk Array [19] with the first row's cover removed [20].

Logical Disk [21]**:** A device that provides an area of usable storage capacity on one or more physical disk drive components in a computer system.

Physical Disk [16]**:** A tangible disk device, such as an internal IDE or SATA hard disk.

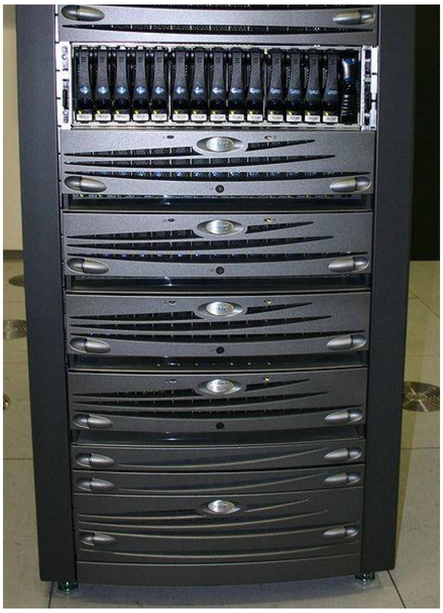
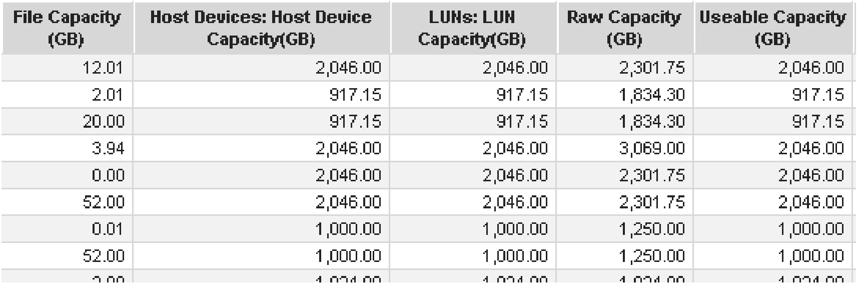


Figure 3-2 One of EMC's Symmetrix Servers [20]

Figure 3-1 also illustrates how these components are connected from a high-level view. For instance, the chain shows how physical disks can be organized into an array, and then broken down to build a thin pool. Then, the chain depicts how these thin pools can be broken into LUNs for the host to view on their virtual machines. This storage is used by the end-users in any way, treated as if the LUN were a part of the computer's hardware, the supply chain masking what happens after the user writes to the LUN.

3.2 Current Solutions

The system that was being improved upon was the Ionix Storage Management program. Its design utilizes column and row tables for SQL Query returns. Because of the massive data blocks and the complexity of the storage chain, the tables usually include plenty of rows and columns, which made the table difficult to interpret. Figure 3-3 displays an example of one of these queries with real data.



File Capacity (GB)	Host Devices: Host Device Capacity(GB)	LUNs: LUN Capacity(GB)	Raw Capacity (GB)	Useable Capacity (GB)
12.01	2,046.00	2,046.00	2,301.75	2,046.00
2.01	917.15	917.15	1,834.30	917.15
20.00	917.15	917.15	1,834.30	917.15
3.94	2,046.00	2,046.00	3,069.00	2,046.00
0.00	2,046.00	2,046.00	2,301.75	2,046.00
52.00	2,046.00	2,046.00	2,301.75	2,046.00
0.01	1,000.00	1,000.00	1,250.00	1,000.00
52.00	1,000.00	1,000.00	1,250.00	1,000.00
2.00	1,024.00	1,024.00	1,024.00	1,024.00

Figure 3-3 Ionix Storage SQL Query Builder [14]

Another way used to demonstrate the architecture of the storage system was to use a complicated topology map, as illustrated in Figure 3-4. However, the graph could get very complex, with increasingly complicated and crisscrossing lines, becoming even more difficult to perceive when there were any more than 20 or 30 devices in one graph.

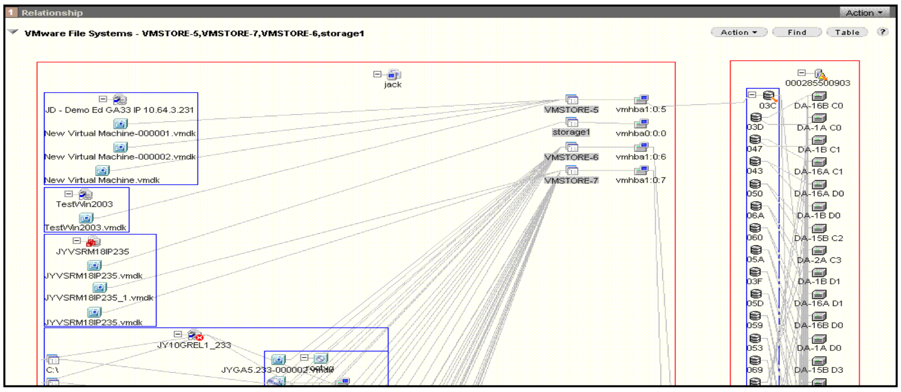
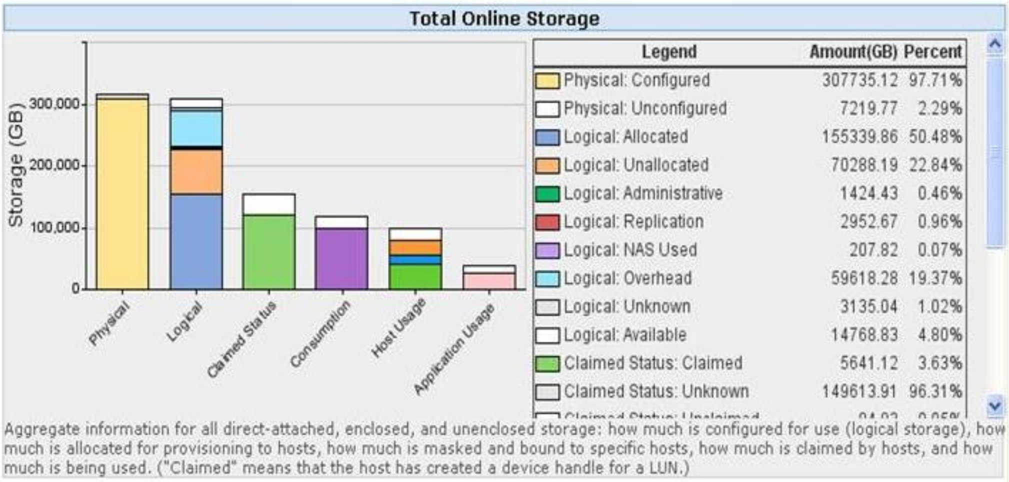


Figure 3-4 Ionix Storage Topology Map Design [14]

Symantec, a leading company that made security software, had a similar storage system to EMC [22]. The visualization they developed incorporated bar charts to display their data, as indicated in Figure 3-5. However, the bar chart shown was complicated and confusing, as there were too many distinct items gathered together in one bar. The color schemas of different items were very similar to each other, which made the display even more difficult to understand.



Legend	Amount(GB)	Percent
🟨 Physical: Configured	307735.12	97.71%
□ Physical: Unconfigured	7219.77	2.29%
🟦 Logical: Allocated	155339.86	50.48%
🟧 Logical: Unallocated	70288.19	22.84%
🟩 Logical: Administrative	1424.43	0.46%
🟥 Logical: Replication	2952.67	0.96%
🟪 Logical: NAS Used	207.82	0.07%
🟦 Logical: Overhead	59618.28	19.37%
□ Logical: Unknown	3135.04	1.02%
□ Logical: Available	14768.83	4.80%
🟩 Claimed Status: Claimed	5641.12	3.63%
□ Claimed Status: Unknown	149613.91	96.31%
□ Claimed Status: Unclaimed	84.83	0.05%

Aggregate information for all direct-attached, enclosed, and unenclosed storage: how much is configured for use (logical storage), how much is allocated for provisioning to hosts, how much is masked and bound to specific hosts, how much is claimed by hosts, and how much is being used. ("Claimed" means that the host has created a device handle for a LUN.)

Figure 3-5 Norton Veritas CommandCenter[22]

4 Design

In constructing the design, there were two questions which were critical to the visualization of the storage. The most important function of a design is to determine what the layout is supposed to accomplish. The difficulty and complexity of the task required that each question that was addressed would likely need to be given its own visual representation within the program. Therefore, each of these questions, also known as use cases, necessitated considerable thought and testing in order for a sufficiently appropriate implementation to be acceptable. The second important function was the conceptual constructs, or abstractions of data, that are necessary for the implementation of these use cases, in order to give an accurate representation of the details. These concepts were critical pieces of the EMC's storage network, and would need to be interwoven with the layout to create a properly descriptive visualization.

4.1 Design Introduction

In total, out of six possible use cases that we were given, three were decided upon to be included in the designing of the visualization prototype for the final product. Each of these use cases targeted a different audience, and required special attention from each member of the group. The variable data depending on what was required at any given point in time was critical to the development.

4.1.1 Use Case 1

The first use case dealt with what a storage and system administrator needed in order to obtain detailed information on the storage system. As mentioned in the background, the Logical Unit Numbers (LUNs) are the closest to a physical SCSI disk that the host sees. For this design to encompass the visualization, the map needed to incorporate multiple details. The first detail included the primary LUNs, or the LUNs which the hosts could interact with as though the LUN was a part of the hard drives attached to the computer. However, as the storage system has multiple backups, the design also had to include the design topology for each of the LUNs. These LUNs could be divided into two groups, depending on purpose. The first group is the Local Replicas, and the second is the Remote Replicas. Since the Local Replicas are immediate replicas on the same system, and Remote Replicas can be located on another array anywhere in the world, the two LUNs would need to be segmented into two distinct groups. This brought the total number of representations in the visualization that would be required to satisfy the use case to three.

4.1.2 Use Case 2

The second use case dealt with the necessity to understand how a thin pool was broken down. To know how the data is being used, or what storage devices are in an array, or where the data is stored, are all valuable questions. To this end, the use case had 5 data types to monitor. The first was Technology, or which types of storage were available. This meant a comparison between SATA drives, Fiber

Channel drives, and Flash drives, among other technologies, are necessary to give a complete analysis of the details. The second data type was Service Levels, based on FAST's tier-based method. The third was Users, which included applications, servers, individual computers, or any other host to which a LUN is subscribed to, and the fourth was Location, detailing where the arrays were located. The final data type was LUNs, and was meant to give information in reference to the primary storage, local replicas, and remote replicas, allowing for a user to more easily break down the first use case for easier viewing.

4.1.3 Use Case 3

The third use case was related to trending and future prediction. In any storage system, there is inevitably a need for growth in order to compensate for the consistent demand for storage space. In a SAN, this can often lead to immense additions to the system being required, but the dilemma used to be solved simply by buying more storage space. With the introduction of thin pools, however, the storage space problem has become smaller, but far more important. Since space can be "over-subscribed", the need for accurate time-keeping has never been more important. Therefore, at almost any stage, on almost any LUN or pool, the time-usage metrics need to be tracked meticulously, as well as recorded for years at a time, as data usage can fluctuate from month to month.

4.1.4 EMC Technology and Use Cases

Out of the use cases, these three were considered to be of the highest priority for needing visualization, as they formed the backbone of the monitoring system that would need to be set up. This necessitated three different designs at the absolute minimum in order to sufficiently fulfill the requirements of the use cases. At the same time, the user needed to be provided with an interactive and engaging simulation without having an overwhelming amount of information, in order to obtain a full and diverse variety of information.

Understanding how the conceptual constructs interweave with the use cases is important to understanding why the use cases are necessary to the development of a system, especially one that can give the necessary details of a complex data set, while still being easy to understand. Each use case interacts with the abstractions, such as thin pools, arrays, and hosts, in different ways, giving different details because of the requirements set for the visualizations. As a result, an understanding of how each use case will be accomplished in reference to the current data structure abstractions is important to understanding how to solve the problem of visualization.

4.1.4.1 Use Case 1 Technology Breakdown

The first use case deals most directly with the top abstractions at their highest level of the LUNs, Thin Pools, Arrays, and RAID groups. The use case has the ability to delve deeper into the backgrounds of each of the abstractions, such as a LUN, but only necessitates this at the highest level. This means that the visualization should be able to give an overview of the entire SAN, from the Host side through the Array side. The visualization can be broken up into multiple layers, but the general idea is to give a “roadmap” to each of the sections that are needed. The system manager would need this quintessential roadmap in order to navigate through the SAN without getting overly confused, and, at the same time, be able to engage the storage manager on a high level in order to create recommendations for future storage ideas and construction practices. Furthermore, the visualization would need to be able to differentiate between the different sections of the SAN, such as the LUNs, Thin Pools, Arrays, and other logical and physical abstractions that exist. This could involve a rather complex legend to explain what each of the icons or “locations” mean, but this would be very helpful to breaking down the SAN.

4.1.4.2 Use Case 2 Technology Breakdown

The second use case is similar to the first insofar as it needs abstractions at the highest level, but since it has to do partly with usage over time, and more with drilling down into deeper strata, the “roadmap” analogy would not apply here. The visualization that needs to be created must have a minimum ability to be able to expand further upon the highest abstractions, without keeping data from becoming obscure or difficult to read, and, at the same time, still keep the groupings that are able to be differentiated from each other with a glance, which may be accomplished by color groupings or another visual means. This visualization needs to be more malleable than that of the first use case, as different datasets need to be viewed depending on what is desired. Location, for example, is not consistent with the Array metric in any understandable way, and must be visualized separately.

4.1.4.3 Use Case 3 Technology Breakdown

Finally, usage over time has a very specific dependency upon time that is not common in the other visualizations. While the second use case could change over time, it would not give a side-by-side comparison with previous iterations and future iterations. This ability is crucial to the future planning of storage additions and tracking data usage of certain groups. While the third use case has wide-reaching implications, from the Array to the Host and even to lower level disks or RAID groups or even throughput over a SAN network, the usage over time should always have a standardized design, so that the visualization does not appear differently with no warning or real understanding of why. If there is a change in the visualization from one view, such as with arrays, to another view, such as the host's data usage, this change should be clearly marked and declared visibly, detailing why this new view is necessary. The user must not be confused, and always be able to intuitively grasp why the changes over time are occurring, and how they can be increased or decreased.

These three use cases and their visualizations will likely not be solved simply by this project. While inroads into possible solutions will be developed and implemented, these visualizations are merely ideas for how to proceed further.

Actual development and deployment of the product in question is beyond the scope of this MQP, and will fall to EMC^2 to continue the work.

4.2 Previous Designs and Their Faults

There were several different methods that have been tried in the past to visualize the complex data that the SAN presents. Each of these has several weaknesses which limit how useful they are in their deployment as visualizations. Each of these have already been present in the product for several years, and are currently considered by the clients of EMC^2 to be sub-par representations of the data, too difficult to get a reliable understanding of the technology available and metrics in use, or simply too obtuse to make any sense given the requirements of the use cases.

4.2.1 Difficulties with Representations

To help understand the problem that these visualizations encountered Figure 4-1 helps to outline the failings of the system and why non-scalable approaches to the problem of visualization are almost impossible to work with. Currently, the internet and data storage devices, on a personal scale, have been around for roughly 20 years. If the data from Figure 4-1 were extrapolated out 10 more years, the data would depict a growth by even more extraordinary factors, and this growth is what must be taken into account as a continual, and possibly exponential, increase. As data is consistently increasing in volume, the necessity of newer and larger storage devices becomes ever more increasing of a dilemma, as well as the backup and storage. The number and complexity of storage systems are unlikely to slow, so any designs must be scalable to match the growth factor that is likely to occur. This means that any system must necessarily be able to handle the most complex amount of data possible in the system.

4.2.2 Bar Graph Representations

The first visualization covered was that of bar graphs. Bar graphs are an industry standard for how to visualize information in a system. There are several problems with bar graphs, most notably the fact that they take up a substantial amount of space while not giving enough details. Considering the number of values that would be required to be viewed at the same time, visualizing the data via a bar graph would result in, at the least, the x-axis becoming crowded or expanding well beyond the range of the screen, and the data varying wildly from exorbitantly large numbers to possibly bytes or kilobytes. The visualization capacity that would be required to accurately display the information for any of the use cases require in a bar graph format would result in far too much data. Even with filtering, the data would be disjointed and difficult to view in light of the whole map. As such, the bar graph is an unsuitable method of visualizing data.

More complicated bar charts, such as Figure 3-5 and Figure 4-1, still suffer from data overflow, with the data being far too difficult to interpret and utilize in an average manner. This problem translates to most forms of bar graphs, therefore resulting in the bar graph being too unwieldy to use effectively.

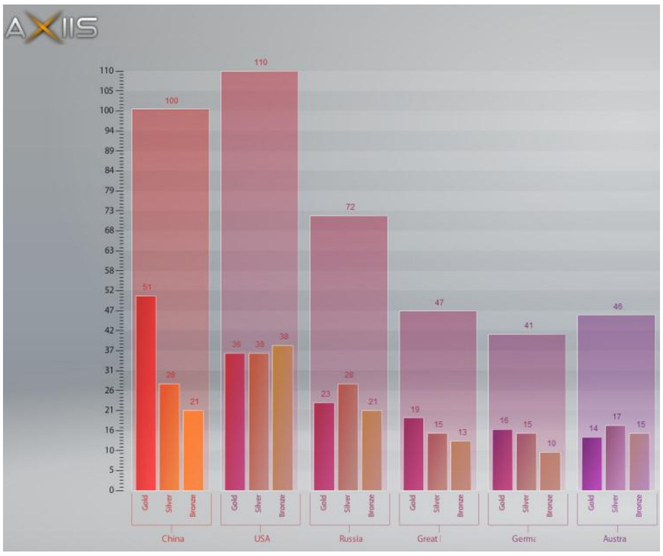


Figure 4-1 Multivariate Bar Chart [23]

4.2.3 Pie Charts

Pie charts were also considered as a method of visualizing the data. This practice of using pie charts in a presentation format has been common in the industry of storage visualization for many years, but the data, similar to the problems with the bar graphs, has resulted in a massive loss of coherence and distinction between the data. For each of the use cases, the pie charts size would need to give visibly distinct and unique colors, a difficult task at best, over a broad range of data, and give appropriate sizes relating to their proportions. The difficulty of this is that humans have significant difficulty in interpreting angles, while differentiating between lengths is substantially easier. This phenomenon is best exemplified in Figure 4-2.

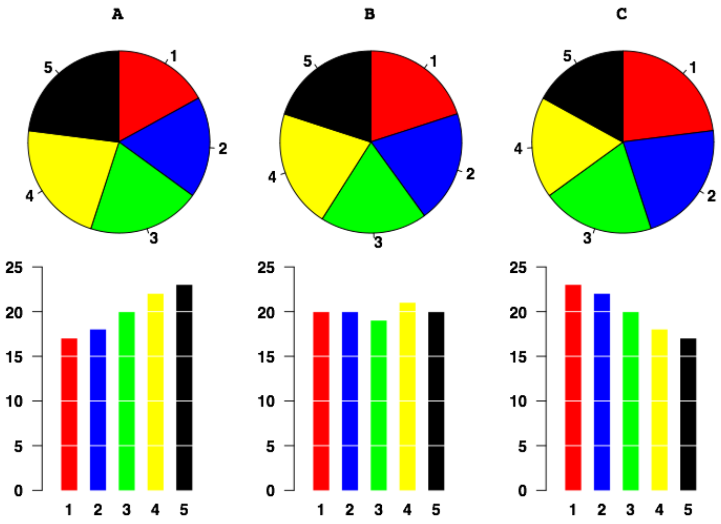


Figure 4-2 Angular Perception Vs. Length Perception [24]

4.2.4 Tables

Another visualization, which is commonly used with EMC's StorageScope program, is a plain table. While not particularly visually appealing, the table is a valid method of visualizing the data in such a way that the user is able to see what is occurring within the dataset in its base form. The problem with this method is that the human mind is not efficient or readily able to understand large blocks of data as a computer can. Because of this problem, tables often feed the users an overwhelming amount of very precise information, without connecting the pieces together that the user is interested in. One of the most important examples of this problem was EMC's SQL Query Builder view that was mentioned in the previous section, Figure 3-3. As was visible in the table, the design was cluttered and bulky, and the relationships between the details were not certain. For example in the File Capacity (GB) column, the returned value is 0.00, but in the Raw Capacity (GB) column of the same row, the returned value is 2,301.75. Due to the confusing nature of the table, and the difficulty to parse visually, the design was not deemed useful for the construction of a comprehensive and easy-to-use visualization.

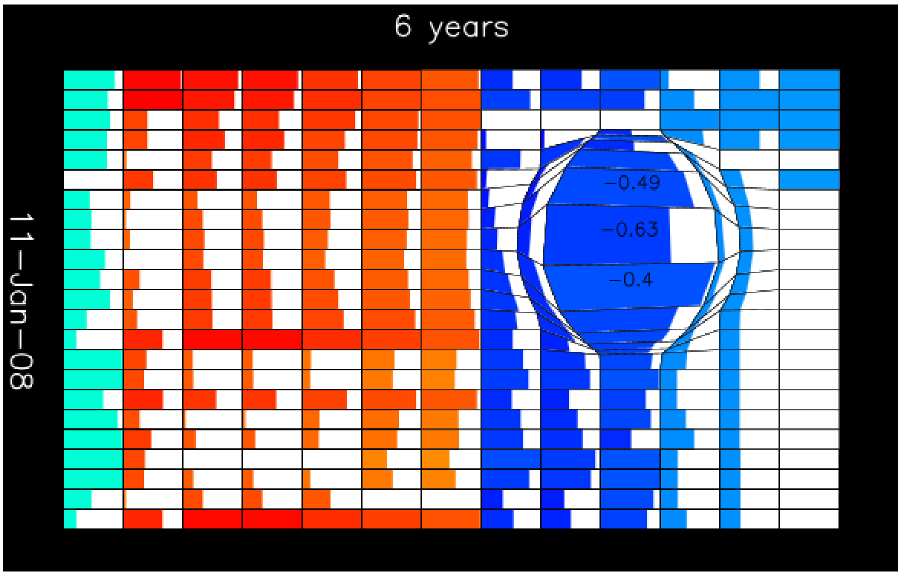


Figure 4-3 Table with Lens and Fill [25]

Another modification of the table design, seen in Figure 4-3, was a table that had a “lens” effect, accompanied by a fill, giving details about the table while displaying a visually appealing design. While more useful than a basic table, the display was still bulky and disorienting, and the distortion effect of the fisheye ended up becoming more difficult to process. The colors in Figure 4-3 also appear to be arbitrary, and while this is not a death knell for the design, it does indicate that the design was difficult to color in a way that signified grouping, other than those that correspond directly with rows or columns, without increasing the strain on the user.

4.2.5 Maps

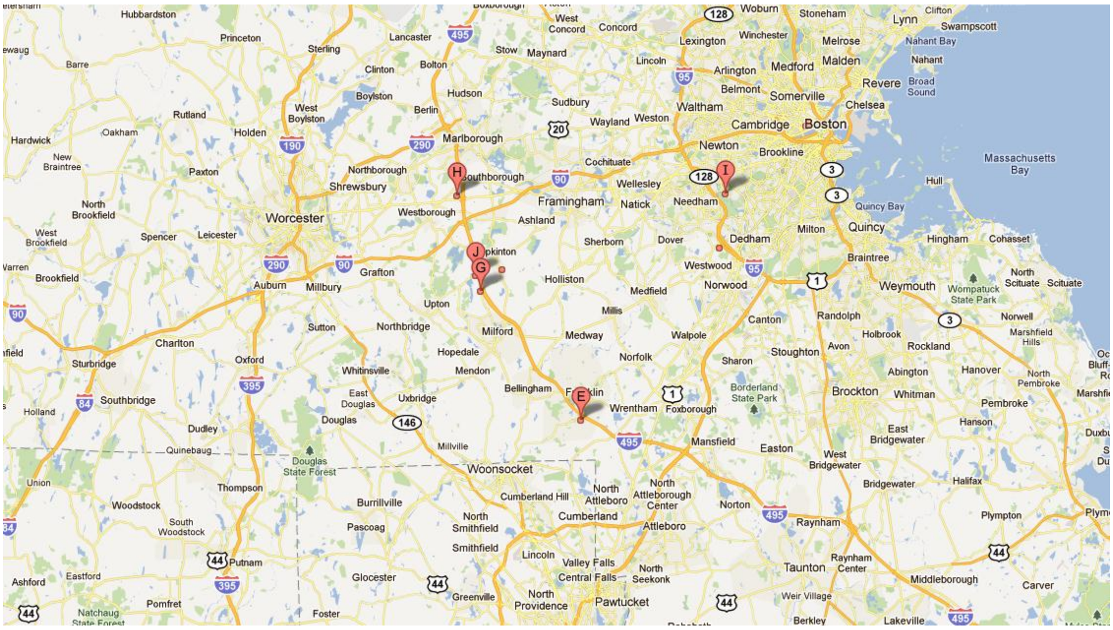


Figure 4-4 Map Design[26]

Since the second use case had Location as one of its required datasets, the idea of a map overlay was introduced, an example of which is Figure 4-4. The problem with a map overlay, even stripping out everything but the locations themselves, was that the map didn't give enough details. If a system manager were looking for the details of a datacenter, they likely would already know where the facility was. The map design was superfluous to the actual project, and could hinder the design as well, since the user's eye was distracted by unnecessary landmarks or geographical features.

Building off of these failed designs, a prototype was drawn up; one that was meant to overcome the failings of previous iterations of these designs, while building upon the strengths of the other visualizations.

4.3 Preliminary Design Concepts

Since each of the three use cases needed to have their own visualization created for their wildly diverse representations of the data available, the project was split into three visualization categories. The first category was tasked with the first use case, and dealt with a representation for explaining to a system administrator why, in the words of Dan Norton, the EMC^2 liaison, "he paid for 250 GB raw, but only received 50 GB of usable space." The second category dealt with the representation of the usage data, differentiated by the 5 different sets of values, and the third category dealt with the representation of the usage over time.

In order to satisfy the first use case, two slightly different ideas were considered for their breadth of details and decisive visualization details. Since each detailed item could necessarily be divided further into thousands of types of data, the highest level needed to be similar to a road map design, showing relationships between the three different types LUNs and their purpose, while explaining which item was currently under inspection. The data should also show a relationship between the LUNs, such that a user could traverse them to get from the beginning of the data chain (at the primary storage) to the end, be they local or remote replicas.

4.3.1 First Use Case Designs

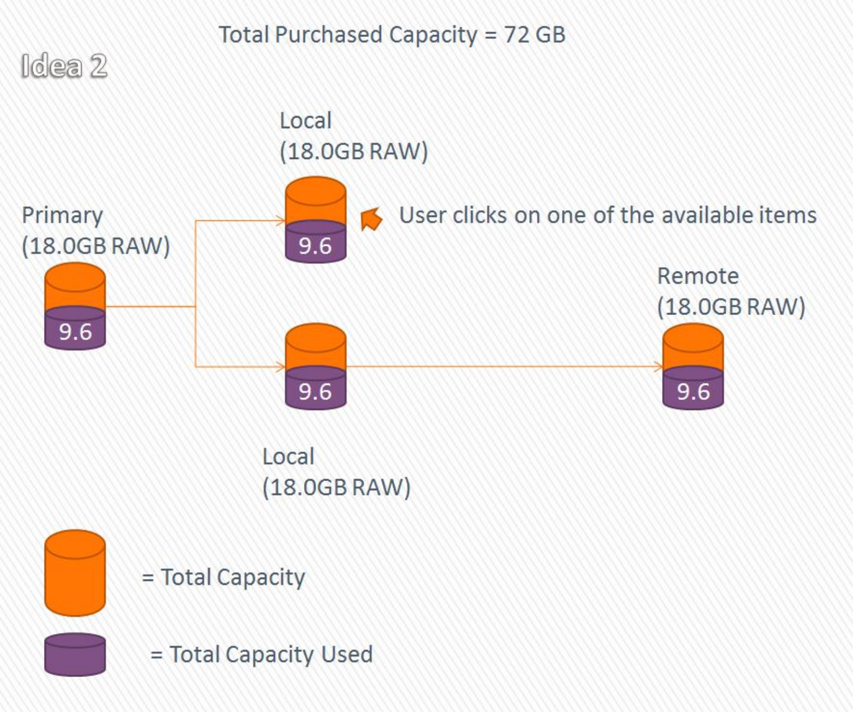


Figure 4-5 First Topology Map Design [27]

The first design, in Figure 4-5, accomplished this by creating a map of the LUNs, placing them on the map in order to give a rough approximation of the chain, as the data moved from left to right. The arrows provided a distinctive method of demonstrating how the data flowed from the input in the primary storage on the left to the final locations on the right. This design allowed the user to easily follow the trail, and it also complied with the F-Read phenomena discussed in recent Human-Computer Interaction (HCI) journals [28], and demonstrated in Figure 4-6. Since the eye, in most English/Romance language cultures, naturally follows from left to right, and top to bottom, a phenomena known as the F-Read Pattern, the necessary information should follow this path as well. The heat map of Figure 4-6 tracks where the eye most commonly rested in the test, with the red signifying more commonly viewed locations, the blue marking less commonly viewed locations, and no color signifying that there was no noticeable attention paid.

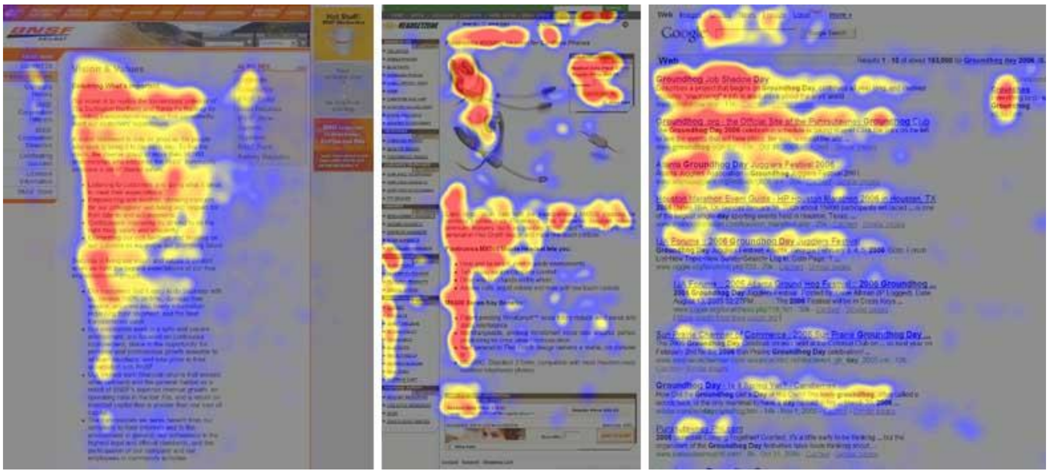


Figure 4-6 F-Shaped Pattern Phenomena Heat Map [28]

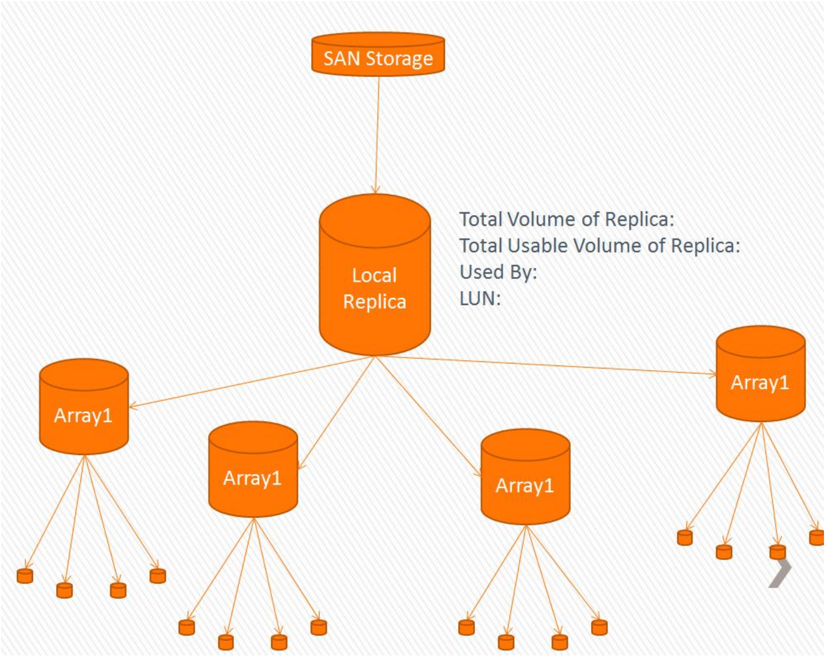


Figure 4-7 Second Web Design [27]

The second design, shown here in Figure 4-7, was also based on the design of the F-Pattern, but this time, in a vertical format. Since one of the benefits of this topology map style was that the lines gave relationships between the nodes, the user can see how a structure relates to another structure. If, for example, someone would like to see all of the structures that are accessible from the primary LUN, then this view would give a web-like design, with accompanying details and a smaller, sub-layer hinted at below the main design. This would permit the user to be able to view the immediate relationships, but also see future relationships. The structure of this design was malleable in that it did not overload the user with too much information, but, at the same time, gave plenty of details relating to the immediate objects the user needed to see and work with.

The third design, which could be an extension of the second design, is in the design of an array. This design was meant to be used in conjunction with the other two visualizations, in order to give a balanced breakdown of the Array, while the others made up the LUN visualization. Since the LUNs follow a different structure to that of the arrays, it is possible for the data, having navigated through the SAN and onto the arrays, to be spread out over many disks. Since the relationship of a SAN's Array to its corresponding Host LUN can be very complex, and the user would likely not be interested in how the data got to that array, only how much of it is stored, the array breakdown would allow better visualization of the array's details, without forcing the visualization to follow the LUN design which is not structured the same way. This design is shown in Figure 4-8.

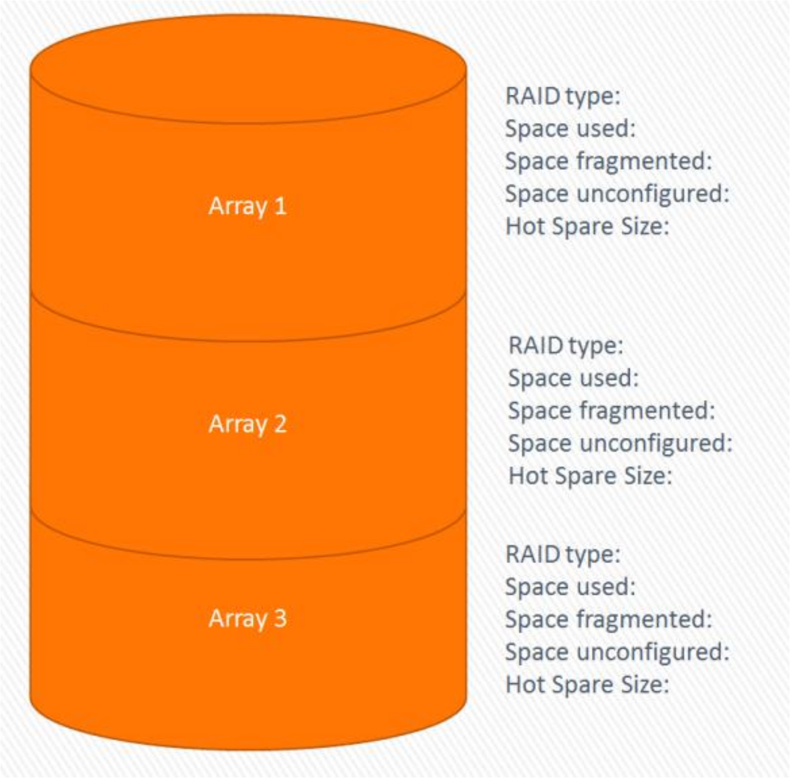


Figure 4-8 Array-Side Design [27]

4.3.2 Second Use Case Design

The second use case was more varied and diverse than the first use case, so a different visualization was decided upon to complete the visualization. The decision rested on a treemap, a modular design that gave a great degree of variability while, at the same time, allowing multiple levels, and, possibly, further drilling into sub-levels (such as going from an Array treemap to a more specific RAID group treemap). This design permitted modifications to the overall size of the visualization, as well. If the host had large volumes of space, the treemap could be made larger. If the host had a smaller amount of space, then the treemap would naturally shrink to mirror this. The available API also had further abilities, such as mouse-over, and alternate views that flowed sequentially from one to the next.

This design allowed a more fluid and visually stimulating setup than was previously available.

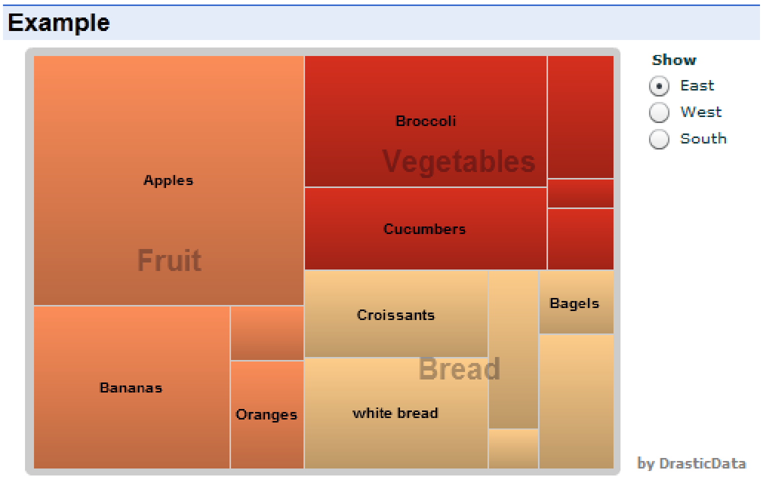


Figure 4-9 Google API Treemap Example [29]

As is visible in Figure 4-9, the treemap also places groups together based on larger details. In this case, fruit, vegetables, and bread all had their own colors associated with their group. If the required item fell under one of those groups, the color scheme would be the same. This allowed the program to give more details than just cursory, high-level information. At the same time, a treemap was complex enough that its options could be modified to fit any of the 5 sets of values that were required of the second use case.

4.3.3 Third Use Case Design

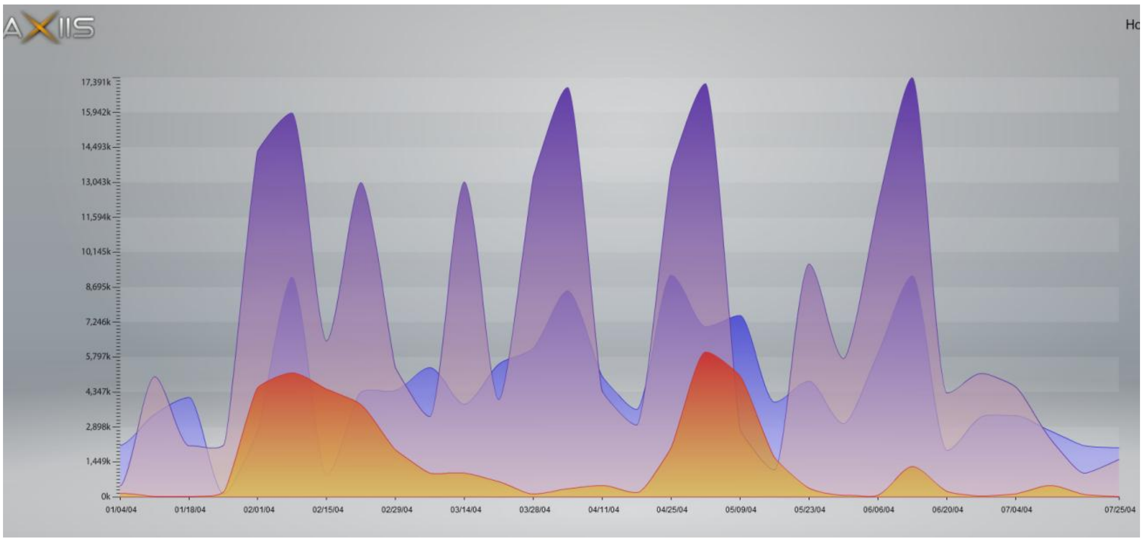


Figure 4-10 Basic Trend Graph Example [30]

The third design, for usage over time, made time an integral part of the structure. In the case of usage over time, the design would have to be able to accommodate both percentages and give actual data. A comparable product, EMC's Pro-Sphere, performed a similar function to what the third use case requires, but did not give understandable percentage visualizations. Therefore, to fulfill the requirements completely, a second visualization was paired with the trend graph Figure 4-10.

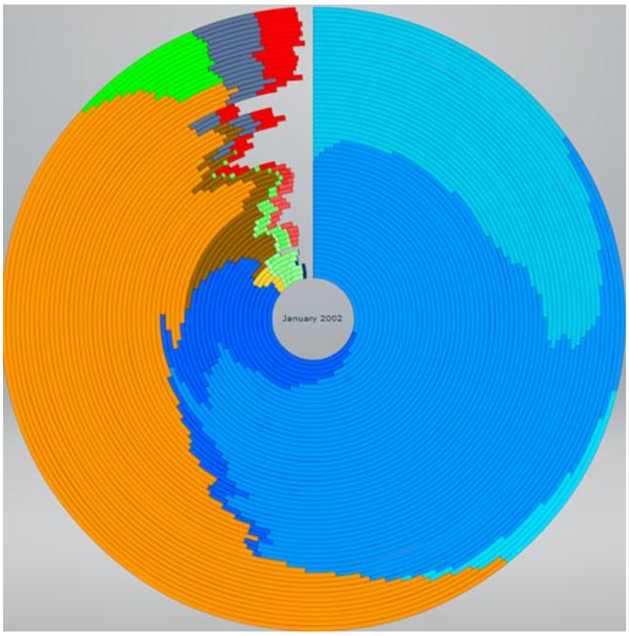


Figure 4-11 Circular Percentage Representation. Depicting Data over Time, with the earliest being the center ring and the newest being the outer ring. [31]

What makes this visualization so appealing is that the model has the benefits of the bar graph, without being overly difficult to read, as shown in Figure 4-11. By creating a circle, the percentage aspect is adequately represented, while plenty of data is detailed to differentiate between one line and the next. The API, owned by Axiis, was also compatible with a mouse-over functionality, which allows for the user to get detailed information on each piece of the circle to inform them of when the chunk was taken, and how much is stored. The design was not perfect; the users, at an informal look at the prototype, still had difficulty understanding the representation, in much the same way that the pie chart was difficult to read through. There are plenty of troubles with the outer and inner rings having identical values, but appearing to be at different points on the circle. However, this model was still an appropriate balance to the trend graph, and gave a focused percentage of the materials available to the host.

4.4 Enhancing the Prototype Design

Having finished the model of what data should be represented in the use cases and how they should be designed to maximize the details available without overwhelming the user, the next step was to enhance the design further, and turn it into a practical design.

4.4.1 User Interface Design

The first problem that needed to be solved was the layout design of the user interface. A promising User Interface (UI) design needs to have a section for navigation and anchoring of data. If the user ever became confused or lost within the layout, then the UI would have failed HCI-compliancy. This problem is difficult for all user interfaces, as there are many details each of the visualizations include that would need to be constrained with a type of border.

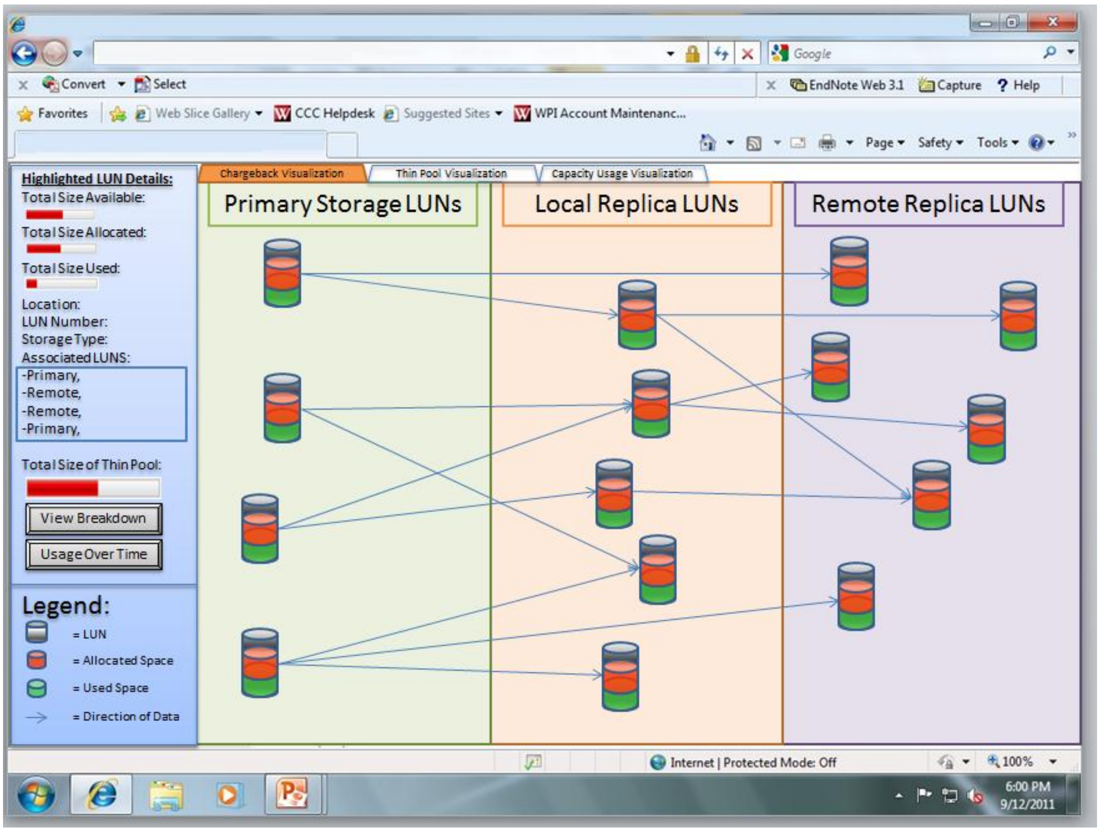
Two designs were created. The first had a horizontal bar at the bottom, which contained all of the information the user needed to navigate, as well as mouse-over details of whatever object the user was looking at. The decision to include mouse-over effects in every aspect of the design was because the user needed to be able to access the data quickly and efficiently, and clicking through multiple layers, or anchoring the cursor/selection to one spot, was slow and clunky. This choice also aided the topology map design for the first use case, as the mouse click could now be used to drag and drop icons within the design around the map. A tabbed-interface was also available at the top of the screen, in order for the user to readily access the most important sections covered by the use cases.

The second design was similar to the first, but had the horizontal bar moved to the left side, in line with the F-Pattern phenomena. After careful consideration, and a preview of the layouts to a select group of EMC employees, the decision was made to use the second design, seen in all pictures of the prototype henceforth.

4.4.2 Topology Map Design Enhancements

While the first use case was satisfied by the topology map, the end result was too complex to be feasible for implementation. Considering that the number of LUNs possible in a single thin pool can be upwards of 64,000, and a regular topology map is not very scalable, several steps needed to be taken to minimize the total size of the map. The first step was a breakdown into two different topology maps.

In breaking down the topology maps, they were redesigned to be more compact, as well as color-coded to differentiate which “item” belonged to which node. In Figure 4-12, shown below, the three sections were divided by their LUNs, to create a “chargeback” visualization, or a visualization which depicts the LUNs relationships. This model was also the first to allow drag-and-drop, in order for the LUNs to be movable. A “distortion” effect similar to the table’s “fisheye” effect mentioned in 4.2.4, would have accomplished a similar effect: the user would be able to examine, in larger detail, the immediate LUN and its neighbors. The necessity of this implementation was due to the issue of scalability. A user needed to be able to see both the larger picture, as well as zoom into the smaller picture. This design proved adequate for small numbers, but to enhance the abilities, the 2D space was mapped onto a 3D cylindrical model, one which allowed the user to spin the cylinder or move left and right, moving to new sections not previously visible in the screen. This allowed the user to "slide" the cylinder to the section that they required.



Highlighted LUN Details:
Total Size Available:
Total Size Allocated:
Total Size Used:
Location:
LUN Number:
Storage Type:
Associated LUNS:
-Primary,
-Remote,
-Remote,
-Primary,
Total Size of Thin Pool:

Legend:
= LUN
= Allocated Space
= Used Space
= Direction of Data

Figure 4-12 Prototype Design, 2D Layout [27]

Since each of the LUNs were able to be dragged, moused over for more information, as well as clicked on to drill down further, the user would see Figure 4-12, a cylindrical representation of the LUN's immediate associations, with the primary, local, and remote LUNs filling in the details. The idea was that, if the user clicked on a LUN at the higher level, the user would like to see only the LUNs immediately available to it. Unrelated LUN information would be stripped away, so as to minimize clutter. This function acted like a filtering algorithm, allowing the user to see only what they wanted to see as shown in Figure 4-13.

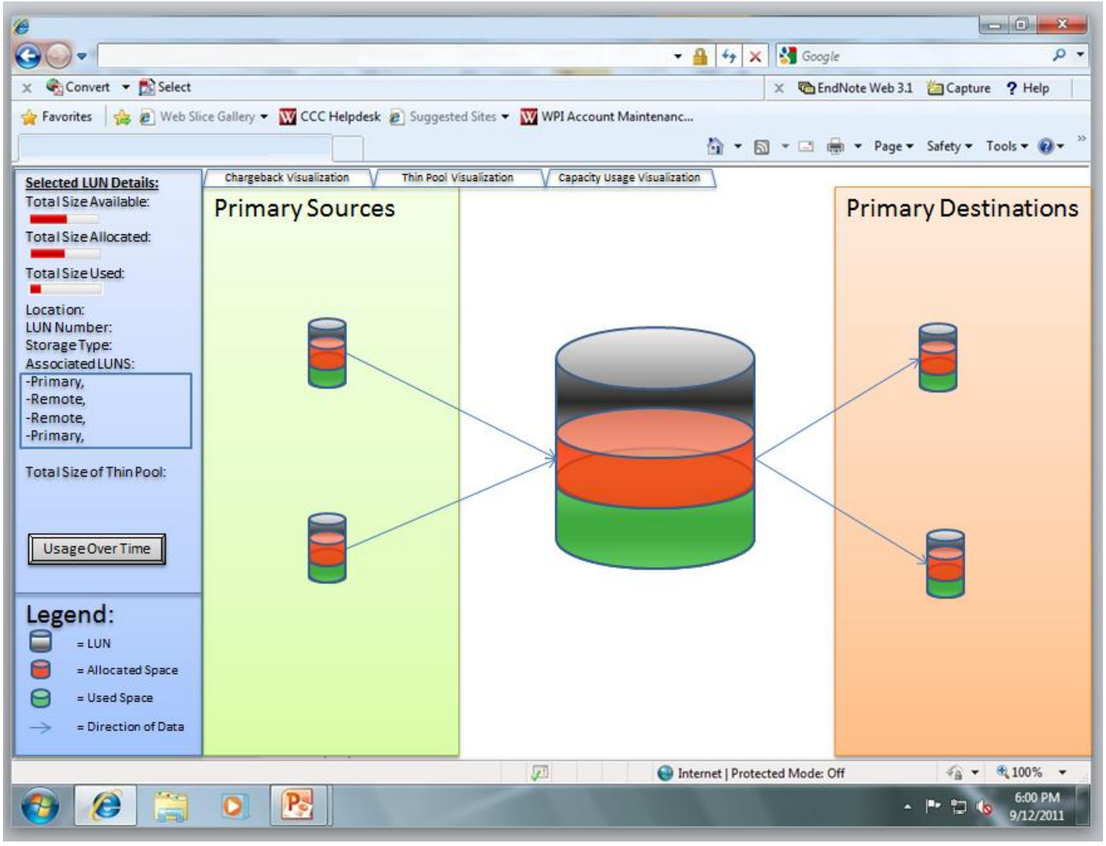


Figure 4-13 Prototype Design, 2D Layout, Second Layer View [27]

The split "second" visualization, which was previously included in the initial version, is the Array visualization. Since each array has thin pools as abstractions of its available space, the array visualization needed to visualize which array the thin pools came from, which thin pool the LUNs came from, and still be able to reach through each layer. If a user clicks on an array, they get a 3D cylinder detailing the array's various disks. Clicking on a thin pool, on the other hand, creates a sphere with the thin pool at the pole, and the LUNs spaced throughout the globe, arranged by their number. Clicking on a LUN, though, leads to the previous visualization in Figure 4-13, allowing a tie-in between the two relationship maps.

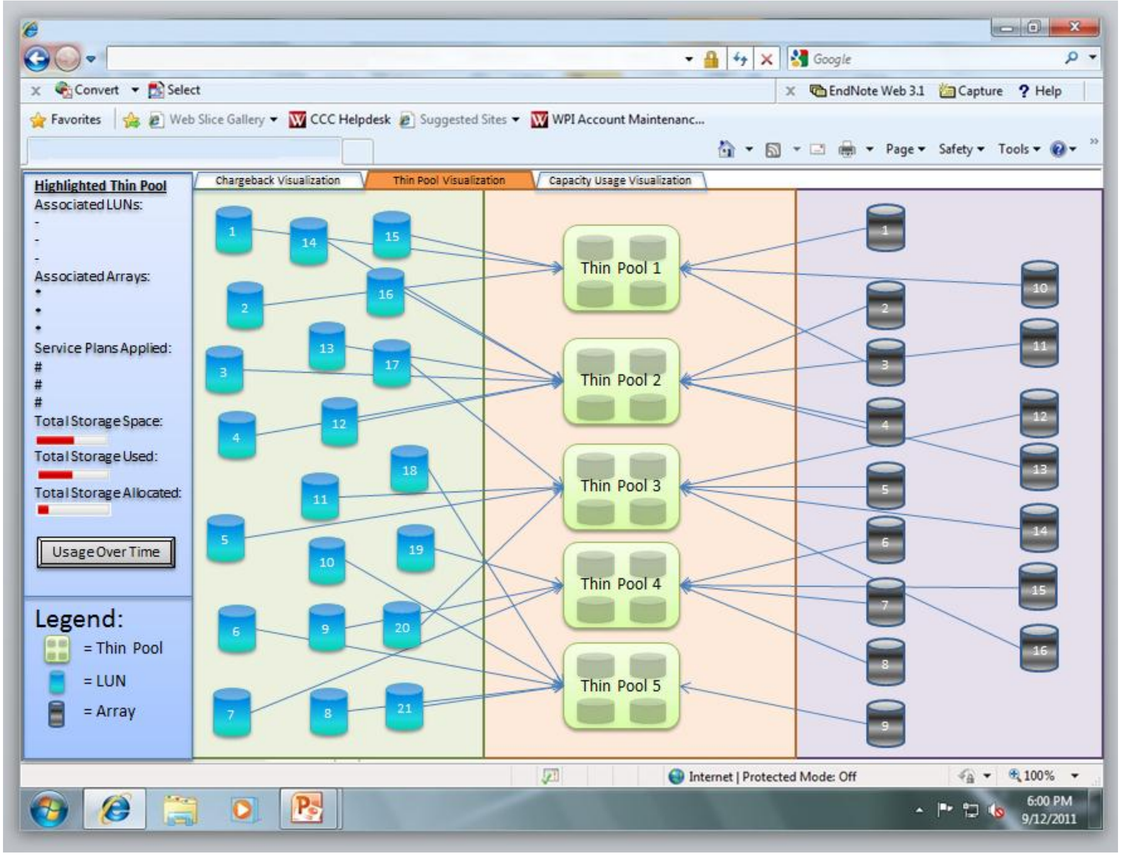
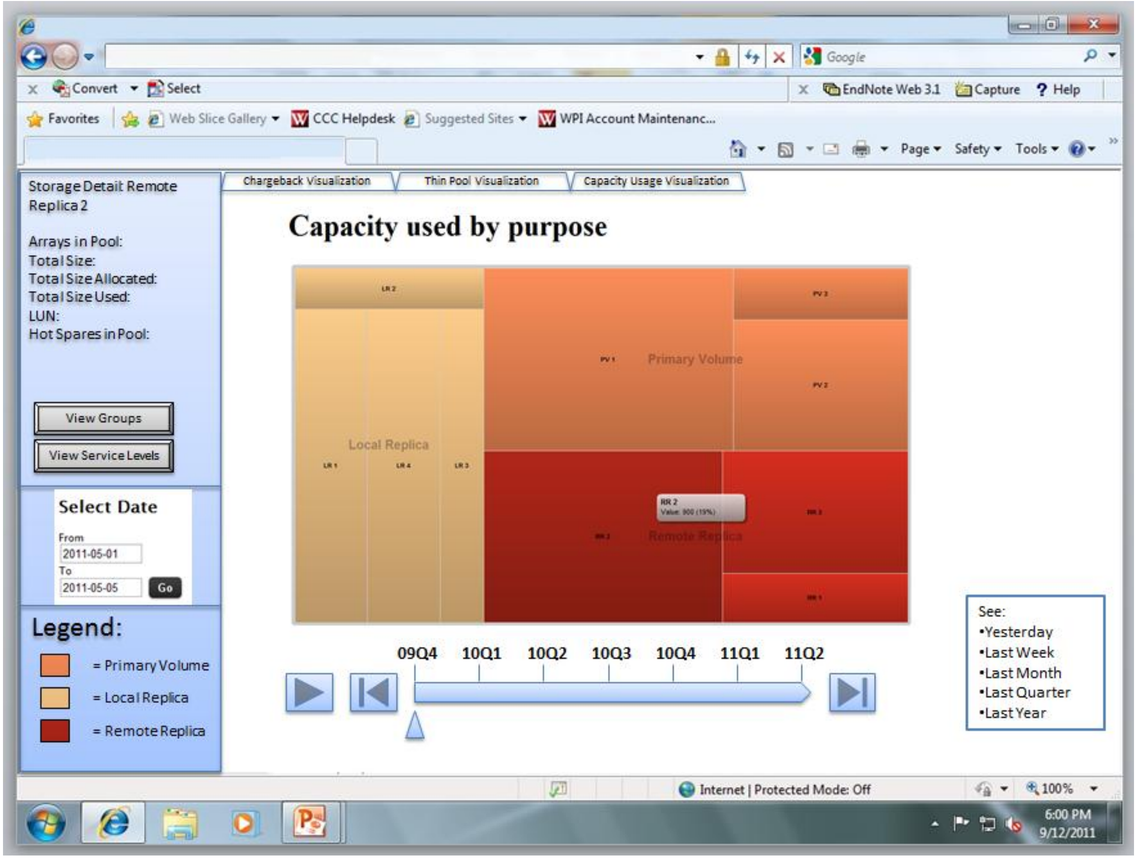


Figure 4-14 Prototype Array Visualization, 2D Layout [27]

A large amount of detail had to go into each of the maps. The array map as shown in Figure 4-14 needed to have a tailored UI on the left side, giving specific details about the disks within, while the thin pool needed to contain information on the used, free, and allocated space. These criteria were critical examples of why the vertical bar on the left side of the screen needed to be both modular and located in a commonly “accessed” space. To an extent, this visualization required using a human’s 3D mental imagination to allow the user to navigate the complex web of the SAN, and minimizing the difficulty by aligning the UI with the F-Pattern reading style proved to be a far more appreciated method of viewing the screen.

A cylinder was chosen for the relationship map because the cylinder is scalable, allowing for the curves to imply infinite “width” around the visualization, and then cutting the cylinder into sections to differentiate between values. A basic 2-D layout was limited by the sides, and therefore unable to cope with the scalable design with accuracy. However, the cylinder could also be bent to “meet” itself again, as some LUNs may connect to themselves. The layout was also more visually appealing than other layouts, which was important due to the necessity to appeal visually to the user.

4.4.3 Treemap Design Enhancements



Storage Detail: Remote Replica 2

Arrays in Pool:
Total Size:
Total Size Allocated:
Total Size Used:
LUN:
Hot Spares in Pool:

Select Date
From 2011-05-01
To 2011-05-05

Legend:
- 🟧 = Primary Volume
- 🟨 = Local Replica
- 🟥 = Remote Replica

Capacity used by purpose

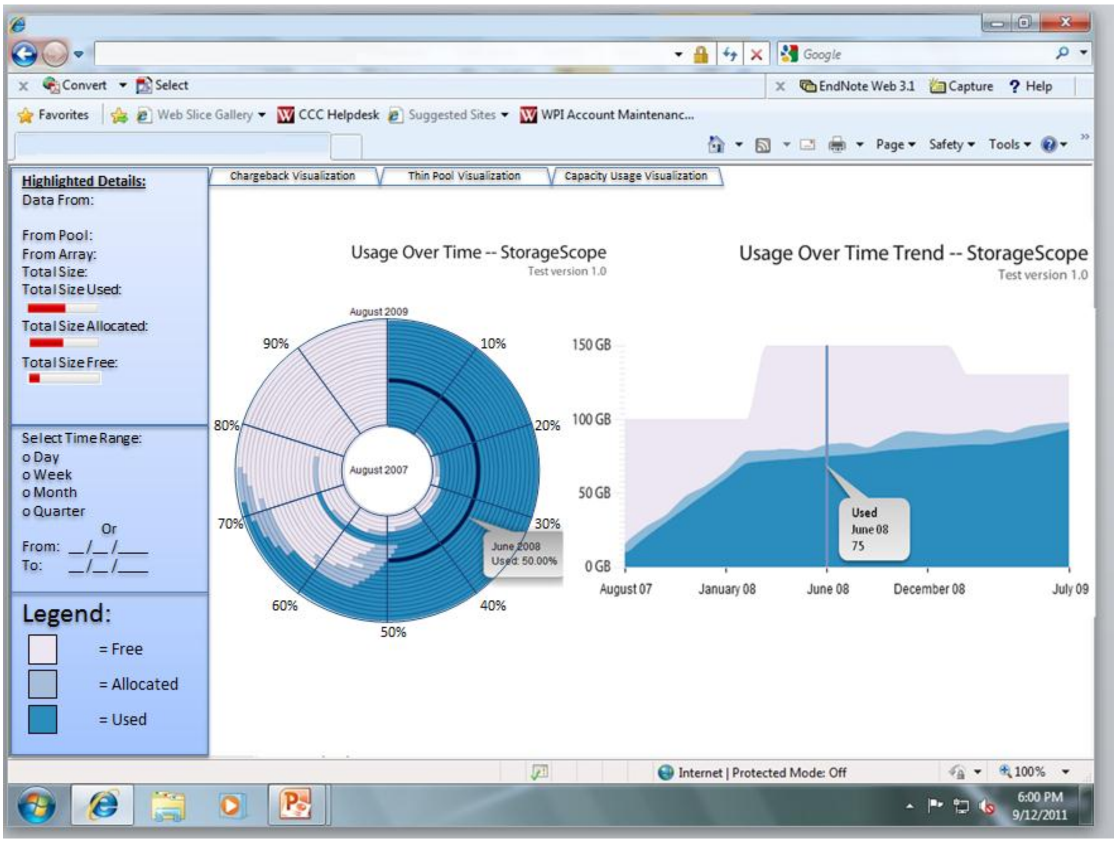
See:
- •Yesterday
- •Last Week
- •Last Month
- •Last Quarter
- •Last Year

Figure 4-15 Prototype Treemap Design [27]

The second use case was satisfied by the treemap in Figure 4-15. This treemap held all of the attributes listed in the previous section, but came with a few additions. The first addition was due to the inclusion of the mouse-over abilities. Since the mouse-over was included ubiquitously in other parts of the program, it was decided that information pertaining to each of the sections that the treemap included would be visible in the left-side bar. This allowed for smaller mouse-over pop-ups as well, which gave vital information, while allowing less-vital information to be represented elsewhere.

The second addition was the inclusion of the slider at the bottom. This slider would allow the user to get a small snapshot of the past few quarters and the "Play" button would allow the user to view the treemap as the data changed over time, partially addressing the third use case as well. This visualization gave a higher level of information than was available in the treemap initially, and allowed for the users to make informed decisions in conjunction with the layout of the third visualization, pictured in Figure 4-16.

4.4.4 Usage over Time Design Enhancements



Highlighted Details:
Data From:

From Pool:
From Array:
Total Size:
Total Size Used:
Total Size Allocated:
Total Size Free:

Select Time Range:
○ Day
○ Week
○ Month
○ Quarter
Or
From: __/__/___
To: __/__/___

Legend:
□ = Free
▨ = Allocated
■ = Used

Figure 4-16 Prototype of the Usage over Time Graphs [27]

The third visualization had some issues that were mentioned in the previous sections, such as difficulty in interpretation due to lack of context, and confusion due to whether the data was increasing linearly over time or not. In order to get around these issues, the design of Figure 4-16 was implemented, in order to fix the problems that were inherent in the designs. First, the details were included in the sidebar in order to give the user more information. Lack of information and overloading the user with information were both critical problems that needed to be addressed in the visualizations, and having the sidebar alleviate the non-critical information overload helped to focus the design. A select few new features were included in order to reinforce the visual aesthetics of the prototype.

The two graphs were linked via the mouse at this point. If the user highlights a bar in the circular percentage graph, the corresponding time in the trend graph would be selected. Similarly, if a time in the trend graph is selected, the circular representation would highlight the ring which represented the corresponding data. The connection between the two helped to impress how the two graphs were intrinsically connected to each other, and that each graph was not to be interpreted individually, but rather as two different representations of the same dataset.

Another feature that was added was the presence of the spider-web design over the circular representation. Since the user had a difficult time reading the angles, the straight lines helped to assuage the issue, as well as give a physical landmark against which the user could "measure" with their eyes. This had the additional advantage of allowing the user to create an additional, user-controlled line, which could be placed anywhere on the circle, acting like a "ruler" for comparison between two bars, generally for those further apart.

Finally, a date-selector was added. This function allowed the user to choose the time period that they were focusing on, as well as break down the information by time intervals, such as days, weeks, quarters, years, or a custom time interval. Once the user had chosen their time interval and period, the graphs would automatically download the data from the SQL database and refresh their displays based on the new requirements.

Figure 4-16's screen was also accessible from each section of the other two visualizations. As the material that the user would like to measure over time could vary, having the graphs be context sensitive allowed the user more variability and control over what they wanted to see.

5 Implementation

The MQP team spent four weeks implementing the prototype demos. The main purpose of the demos was to provide the EMC engineers a better understanding of the ideas the MQP team had proposed. The following technologies were used in the implementation: HTML5 [32], Google Chart APIs, Drastic Treemap, jQuery, Adobe Flex and Axiis library. This section will explained how the team used the listed technologies to implement the demos and the difficulties encountered during the implementation.

5.1 Chargeback & Array Visualization

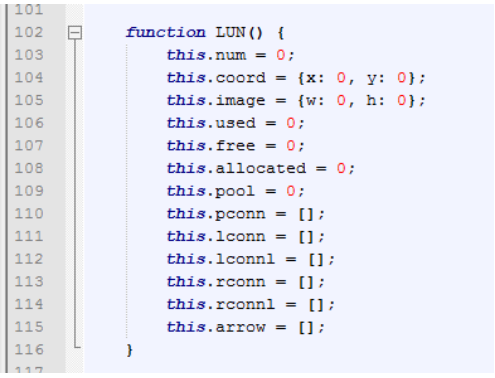
The first use case was difficult to visualize without the design of its own interface due to the complex demands of the visualization, the requirement of scalability, and the necessity for the design to be modular. Once the interface was designed, the implementation became the most important item to complete, as the evaluations would need a functional design in order to be accurately represented. The chargeback/array visualization was rendered in HTML5, using the <canvas> element [33], and this required that the visualization comprise both a UI design for detailing the information, on top of the visualization itself. The design was eventually split into two "state" groupings, with states 0 (primary LUNs focused on), 1 (local LUNs focused), and 2 (remote replicas focused) referring to the chargeback visualizations representation of the data, and states 3 (RAIDs focused), 4 (Arrays focused), 5 (Thin Pools focused), and 6 (LUNs of Thin Pools focused) referring to the Array visualizations.

5.1.1 Technology – HTML 5 and Javascript Code

The choice to use HTML5 (with Javascript backing up the majority of the complex scripting parameters) was made very early on, because HTML5 and Javascript, when combined, lend themselves more to rapid development and modification than most other programming languages. If the program had been coded in Java or another language, the integration of the program into a friendly interface, much less a fully functional UI, would have taken too much time away from the project's goal of constructing the visualization. As is visible in Figure 5-1, the <canvas> element is a powerful tool that allows drawing, while still being completely integrated with HTML elements, allowing for a Javascript program to easily transfer information from the canvas element into the browser without difficulty in design.

Figure 5-1 HTML5 Canvas Demonstration with Javascript Modifiers in Upper Right [34]

All information that was drawn on the canvas was stored as Javascript elements on the page in question. To keep the HTML page from becoming too cluttered, the Javascript code was stored in a separate file (called "visualization.js"), which created random data, parsed it, placed it on the <canvas>, and incorporated it into the UI available on the main page. The code for the element "LUN," which was a consistent representation from both the chargeback and array visualizations, is demonstrated in Figure 5-2.



```
function LUN() {
    this.num = 0;
    this.coord = {x: 0, y: 0};
    this.image = {w: 0, h: 0};
    this.used = 0;
    this.free = 0;
    this.allocated = 0;
    this.pool = 0;
    this.pconn = [];
    this.lconn = [];
    this.lconnl = [];
    this.rconn = [];
    this.rconnl = [];
    this.arrow = [];
}
```

Figure 5-2 Sample Code of LUN Structure [35]

This code used a basic template that was common to all of the structures that represented data, each of which was stored in an array of its own, so as to maintain data independence and simplify access of non-related structures. Since each was an image representing their respective structure, the relationship “arrows” that attached to it needed to know the image’s height, width, x and y coordinates, parent node, child node, and its designation. The “num” element specifically referred to the “number” of the LUN, since the designation and its location in the array could be different. However, for simplicity sake of the program, array location was used to locate a specific item in an array. Additionally, since the coordinate geometry of an HTML5 <canvas> element positions the upper left corner as (0,0), and increases along the corresponding axis[36], the image location and size parameters were used to calculate where the arrow should be located, and where the picture should be displayed. If an object was above a previous one, then the y offset for the lower object would need to be half the image’s height, displaced negatively (which would move the translated point up on the <canvas> element), and the y offset of the other would be half of its image's height, displaced positively. The canvas (x,y) numbering schema can be seen in Figure 5-3.

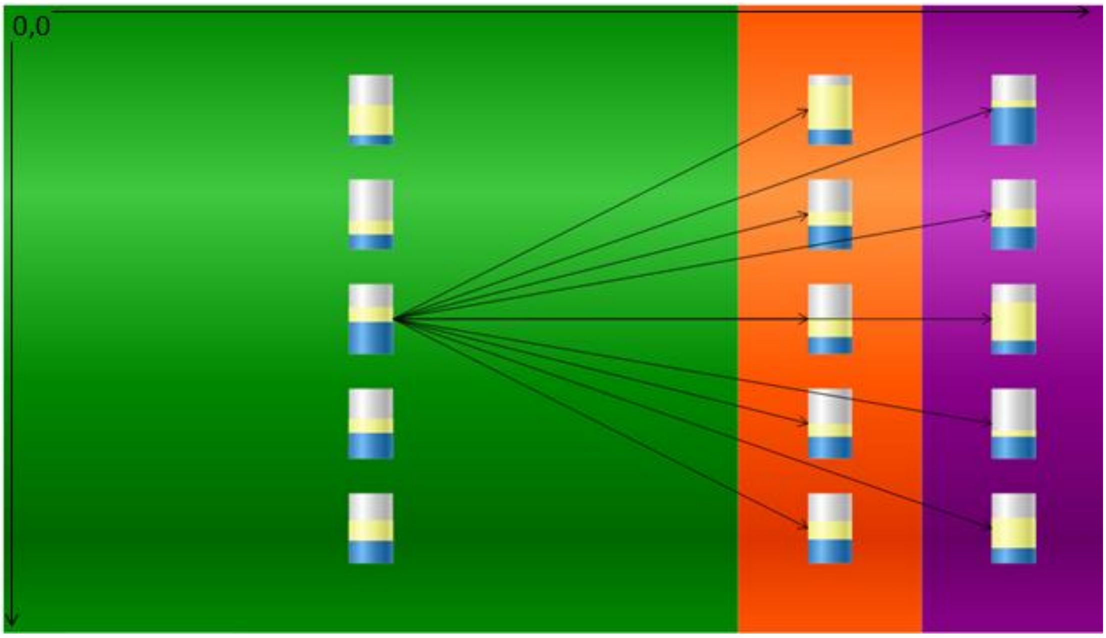


Figure 5-3 Canvas Element Example with Chargeback Visualization, State 0 [35]

The interconnectivity of the HTML5 and the Javascript, operating in an object-oriented language while using the "setInterval" command to monitor user input, allowed these structures to be utilized fully to create a visually pleasing replica of the designs that had been developed in the previous section, while still responding to user commands quickly, preserving the user's sense of control over the environment.

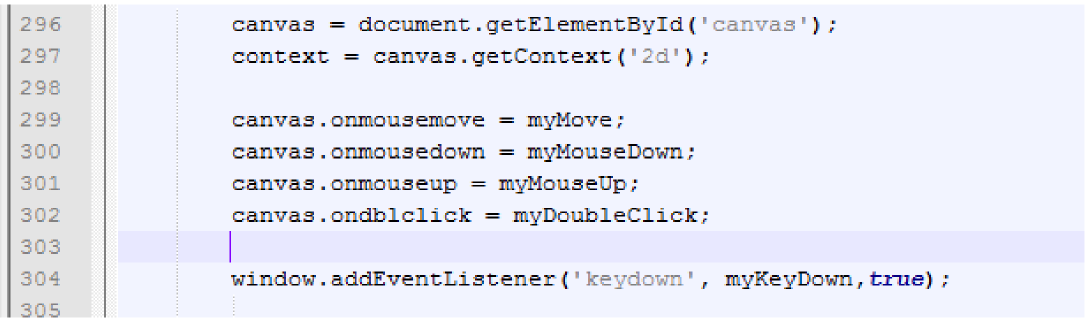
However, HTML5 does have several drawbacks. Since the new features, such as <canvas>, are still in their infancy, the program is not powerful enough to include all of the features that are anticipated in the final version. One of these features is a 3D <canvas>, which would have permitted the full realization of the HTML elements.

5.1.2 UI Design Implementation

The design for the UI was more difficult to implement than was anticipated. The main requirements of the UI was a mouse-over functionality, a breadcrumb "tracking" system giving the user the ability to navigate through the system based on previous choices, a method to navigate possibly thousands of LUNs quickly and efficiently, scalability in its development, and animations, all without running into the current limitations of the HTML5 canvas element

5.1.2.1 Mouse Controls

Implementing mouse controls on the canvas is not a difficult task. The <canvas> element already comes with several options for integrating the mouse and the keyboard into the element. Each of these options are readily available to create with the code displayed in Figure 5-4, and canvas stored as a global variable.

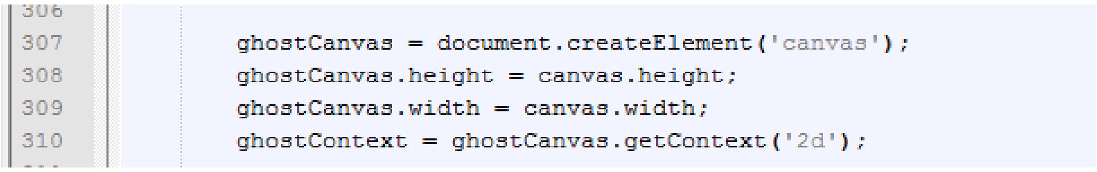


```
        canvas = document.getElementById('canvas');
        context = canvas.getContext('2d');

        canvas.onmousemove = myMove;
        canvas.onmousedown = myMouseDown;
        canvas.onmouseup = myMouseUp;
        canvas.ondblclick = myDoubleClick;

        window.addEventListener('keydown', myKeyDown,true);

```

Figure 5-4 Canvas Setup [35]

The mouse-over functionality, represented by "canvas.onmousemove", was perhaps the most difficult of the materials in the implementation to correct. As the

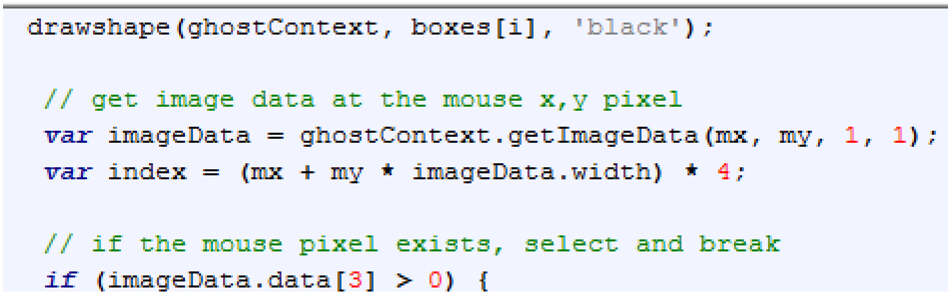
<canvas> element does not store its current status, and renders everything in a "pixel" basis, picking out a single element on the canvas is impossible. A pixel, by definition of the canvas, is either colored or not. In order to get around this technical hurdle, a "ghost canvas" was implemented. The code for the ghost canvas is shown in Figure 5-5.



```
ghostCanvas = document.createElement('canvas');
ghostCanvas.height = canvas.height;
ghostCanvas.width = canvas.width;
ghostContext = ghostCanvas.getContext('2d');
```

Figure 5-5 Ghost Canvas Element [35]

This element is a global variable that creates an element in the DOM structure, without attaching it to the webpage. This has the effect of storing an undrawn "canvas" to paint structures on in memory, without including either background colors or unnecessary structures. This way, a simple function to check if the selected pixel is painted can be implemented in the mouse-over, allowing the user to quickly and easily select on object without unnecessary computations reducing the speed of execution. This code is visible in Figure 5-6.



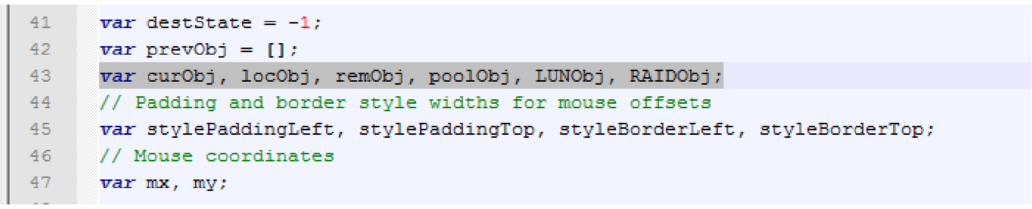
```
drawshape(ghostContext, boxes[i], 'black');

 // get image data at the mouse x,y pixel
 var imageData = ghostContext.getImageData(mx, my, 1, 1);
 var index = (mx + my * imageData.width) * 4;

 // if the mouse pixel exists, select and break
 if (imageData.data[3] > 0) {
```

Figure 5-6 Ghost Canvas Pixel Test [35]

Furthermore, by tracking the painted pixels, drawing a full image is not necessary, and can be replaced by an invisible, black box, drawn only when not “animating” the objects onscreen. This speeds up computation time for drawing the ghost canvas while still aiding the display of the main canvas.

5.1.2.2 Breadcrumbs

The term “breadcrumbs” refers to a method of coding that tracks previous elements, allowing the user to get to the location they previously had, without difficulty in transitions. In order for this to be successfully implemented, each state’s object that the user passes through needs to be saved as a separate construct. In the case of this code, each state saved the currently selected object (be it LUN, Array, Disk, or other element) into a variable to allow the user to backtrack. Certain mouse clicks could change these saved objects, such as a user in a local LUN state clicking on a different primary LUN than the stored primary, which would result in a change in the overall view. Figure 5-7 demonstrates how this was implemented.



```
var destState = -1;
var prevObj = [];
var curObj, locObj, remObj, poolObj, LUNObj, RAIDObj;
// Padding and border style widths for mouse offsets
var stylePaddingLeft, stylePaddingTop, styleBorderLeft, styleBorderTop;
// Mouse coordinates
var mx, my;
```

Figure 5-7 Global Variables Relating to Breadcrumb Design [35]

Since states 0 and 4 were mutually exclusive, curObj was shared between the arrays and the primary LUNs of the two fields.

5.1.2.3 Rapid Navigation

As the user needed to navigate rapidly through what could be thousands, or even hundreds of thousands of LUNs, the option to allow the user to “select” which of the objects they wanted to look at was developed. The code is a small HTML form placed below the canvas, which, when a state is selected, will contain a number stating which LUN is currently selected, and the total number of LUNs which are available in the system. This allowed the user to change the value that they were looking at, and move to a different section of the state easily.

5.1.2.4 Scalability

One of the largest issues that the use cases had was scalability. As a LUN could be enormously complicated, with dozens of backups, and even backups of the backups, there needed to be a scalable design. This was solved with the presentation of the cylinder. Each of the sections that were covered had the possibility for expansion. In order for the scalability of the design to be realized, both ends could be added to, allowing for the “creation” of new slices of the cylinder, which could host another set of data structures.

5.1.2.5 Background Sectioning

In order to section the backgrounds properly, they were represented as rectangles with a gradient at pre-determined locations along the LUN. Figure 5-8 shows this code in more detail, but the sectioning was varied based on what state the user was in. For states 0-2 (primary, local, and remote replicas), the user would see the width change. States 3-6 would vary in height and location along the canvas, though. This dynamic representation allowed animations to be implemented easily as well.

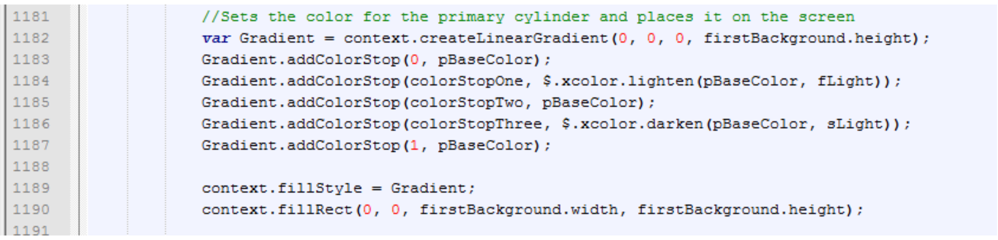
5.1.3 Chargeback Implementation

Having successfully completed the implementation for the UI, the next step was to flesh out the program with the actual layout and placement of objects. Up until this point, nothing had been drawn on the canvas element. All programming had been checked via a series of test cases which were replaced with complete code schemas.

One example of these test cases was a "temp" code which verified the status of the LUNs in their random creation, since extracting the data from the SQL database we were provided with was nearly impossible. This test case verified that each "LUN" created did not exceed the size of the thin pool as a whole, which in turn did not exceed the size of the array, which in turn equaled the total size of the included RAID groups, which in turn equaled the total size of the included disk groups. This representation was difficult to procure, but successfully illustrated the nature of the LUNs.

The vertical gradient, in terms of code, was easy to create, as the gradient is a piece of code specifically designed for representing 3D objects without actually being in 3D[37]. However, the problem lay in sizing it in accordance with the screen. As the user's screen can change width and height, depending on how the user would like to view the representation, the variable width and height of the browser at all times would need to be taken into account.

This was accomplished by setting values according to their height of their structure. In the case of Figure 5-8, the values of the gradient were changed depending on the height of the background, giving a uniform design layout to the cylinder, and allowing the user to view a smooth animation when the change in height occurred. Also visible in the Figure 5-8 is the “context.fillRect” code which gives the size of the background section, which, in this case, is variable to allow for animation changes.



```
//Sets the color for the primary cylinder and places it on the screen
var Gradient = context.createLinearGradient(0, 0, 0, firstBackground.height);
Gradient.addColorStop(0, pBaseColor);
Gradient.addColorStop(colorStopOne, $.xcolor.lighten(pBaseColor, fLight));
Gradient.addColorStop(colorStopTwo, pBaseColor);
Gradient.addColorStop(colorStopThree, $.xcolor.darken(pBaseColor, sLight));
Gradient.addColorStop(1, pBaseColor);

context.fillStyle = Gradient;
context.fillRect(0, 0, firstBackground.width, firstBackground.height);
```

Figure 5-8 Sample Background Creation Code [35]

To further the illusion that the background was also curved, a minor “distortion” effect was implemented to make the objects increase in size the closer they were to the focus of the state. This functionality required the use of the context.scale function, which allowed the user to increase the size of the objects they were creating. This also displaced the arrow creation algorithm, which required a new design as a result.

5.1.4 Array Implementation

Unlike the Chargeback implementation, the array implementation focused on the central area over the side areas. While this had the result of using similar code as the chargeback visualization, there were a few changes that needed to be made.

The first change was that the state would need to be focused in the center, vertically instead of horizontally. While the chargeback visualization always maintained three visible states of varying lengths to display what was focused, the cylinder was easy to implement because the gradient was uninterrupted down. The design for the array visualization, however, forced the gradients to be marked differently. As the screen was built down, the colorStops located in Figure 5-8 were modified to higher values, in order for the gradient to appear as though it became light in the first state, increased and then decreased in lighting in the second state, and finally decreased considerably in the third state.

5.2 Capacity Usage Breakdown Visualization

The second use case was tackled by using an interactive treemap. For instance, the treemap breaks down the total capacity based on locations and consumption status (namely ‘used’, ‘free’, ‘allocated’). Users interact with the treemap by a single-click or double-click on different sections of the treemap. A bar graph on the left panel of the screen, showing the current usage statistics, and a series of sparkline images, showing the trend of usage over time, give more information to the users. Figure 5-9 is an annotated screenshot.

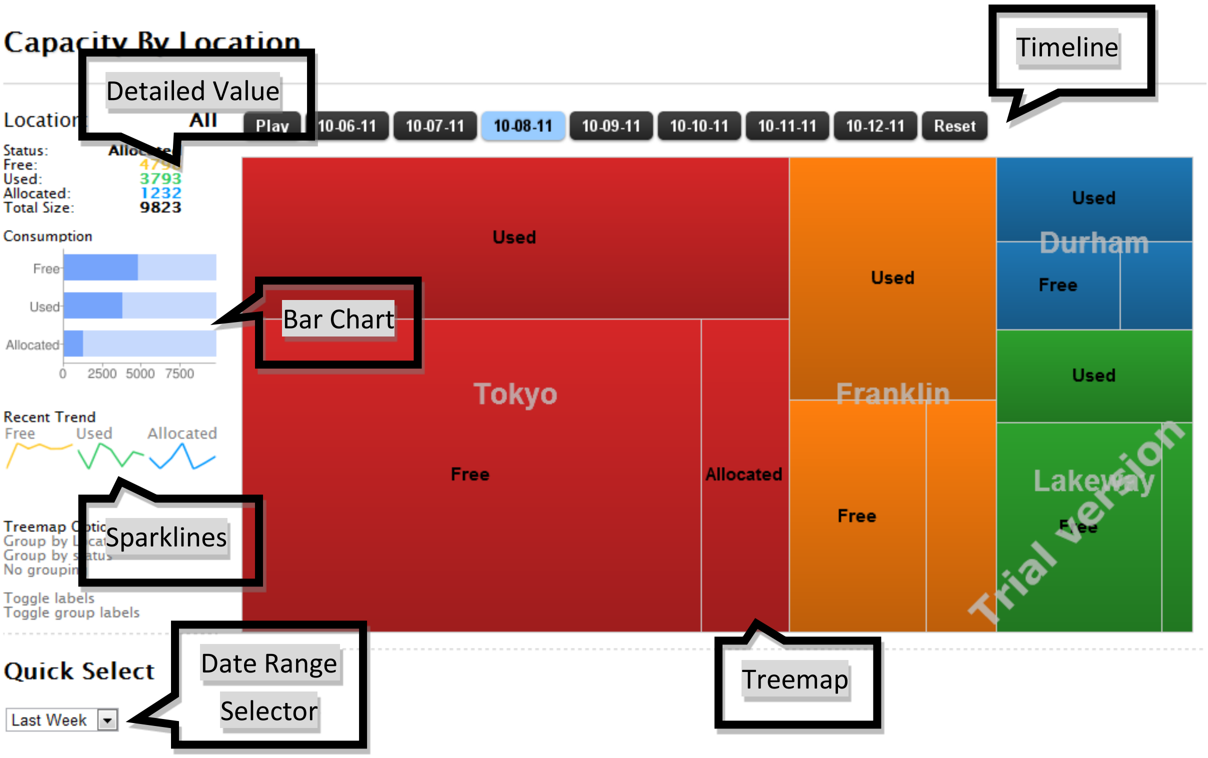


Figure 5-9 Annotated User Interface Explained [35]

5.2.1 Some Major Technologies

To implement this part of the prototype, a few existing tools and technologies were used and modified. For instance, some technology, such as Drastic Data Treemap [38] and Google Chart API [39], were used to render visualization charts. Some tools, namely JavaScript and jQuery Tools, were utilized to create an interactive user interface. Other technology, such as Hypertext Preprocessor (PHP) [40], was used as a backend tool in order to interact with database.

5.2.1.1 DrasticTreemap

As mentioned in section 2.1, treemap is able to display hierarchical (tree-structured) data using a set of nested rectangles.

DrasticTreemap is a dynamic treemap built in flash by DrasticData [38]. A treemap is a method for displaying tree-structured data using nested rectangles, see Figure 5-10. Most treemaps tools on the internet are static. However, DrasticTreemap is dynamic in the sense that it smoothly resizes and repositions its items to display a new state. The dynamic transition is very useful because it gives the user a good idea of how size changes. With smooth animation, it is very easy to spot the changes in size. That was one of the major reasons that it was chosen as a tool for the visualization. In addition, the MQP team added the functionality that the total size of the treemap will change based on the actual total value. This way, not only can the users see how different element compares relative to each other, they can also easily tell how much one element grow or shrink in terms of absolute values.

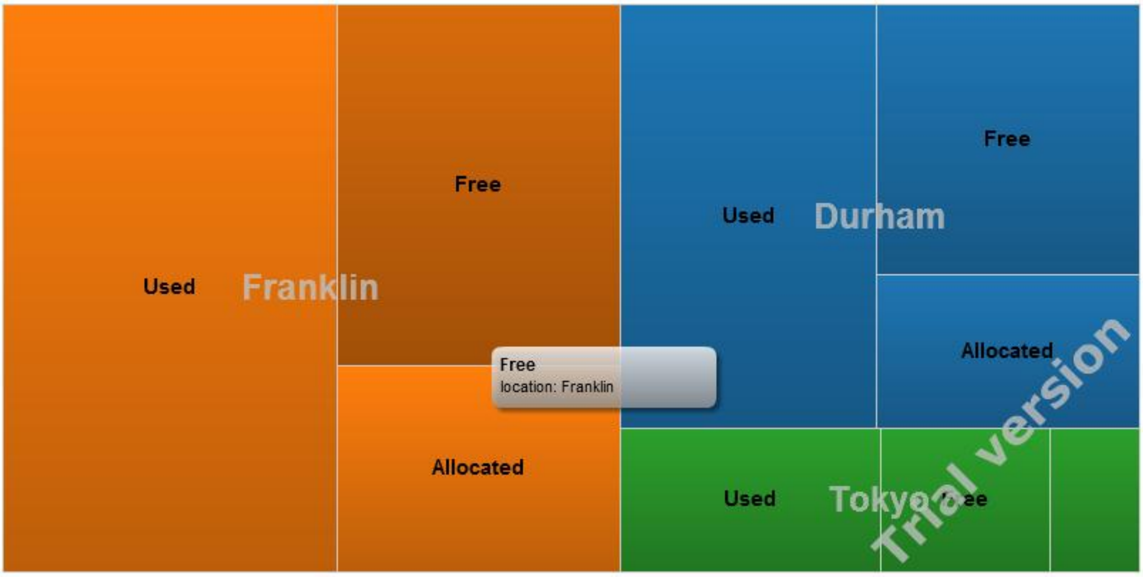
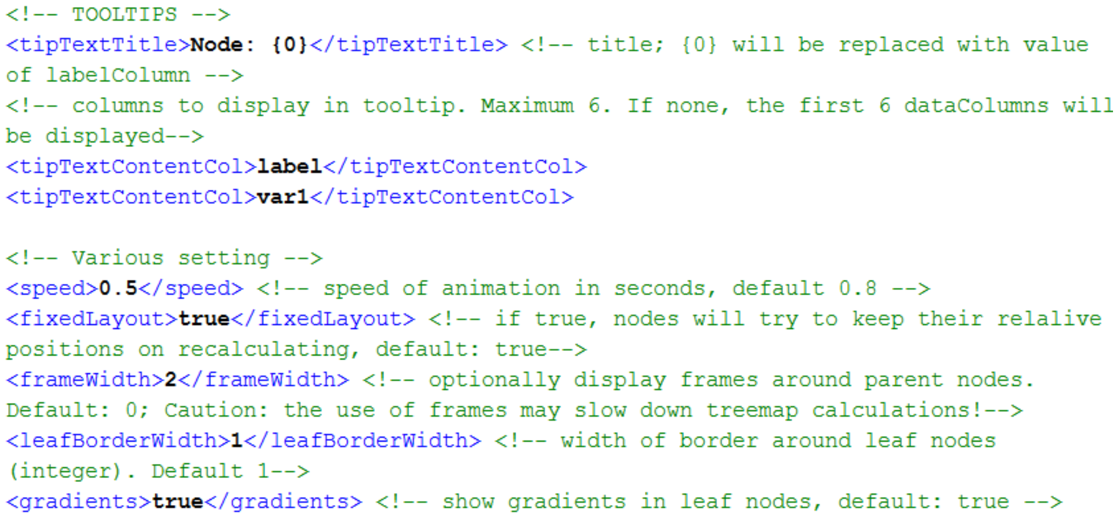


Figure 5-10 A Sample Treemap [35]

The treemap exists as a Flash object in a webpage. In order to add it into the webpage, SWFObject was used to embed the Flash object [41]. SWFObject is a commonly used standards-friendly method [41] that utilizes a small JavaScript file.

One of the most important files in DrasticTreemap was the config.xml file. It defines some properties (namely color, animation speed and text size) and behavior (namely mouse-over effects) of the treemap, see Figure 5-11. In this MQP, this configuration file was automatically generated by a PHP script at run time. This way, the treemap was more flexible because it could read data of different sizes.



```
<!-- TOOLTIPS -->
<tipTextTitle>Node: {0}</tipTextTitle> <!-- title; {0} will be replaced with value
of labelColumn -->
<!-- columns to display in tooltip. Maximum 6. If none, the first 6 dataColumns will
be displayed-->
<tipTextContentCol>label</tipTextContentCol>
<tipTextContentCol>var1</tipTextContentCol>

<!-- Various setting -->
<speed>0.5</speed> <!-- speed of animation in seconds, default 0.8 -->
<fixedLayout>true</fixedLayout> <!-- if true, nodes will try to keep their relalive
positions on recalculating, default: true-->
<frameWidth>2</frameWidth> <!-- optionally display frames around parent nodes.
Default: 0; Caution: the use of frames may slow down treemap calculations!-->
<leafBorderWidth>1</leafBorderWidth> <!-- width of border around leaf nodes
(integer). Default 1-->
<gradients>true</gradients> <!-- show gradients in leaf nodes, default: true -->
```

Figure 5-11 Part of the Config.XML [35]

5.2.1.2 Google Chart API

The Google Chart API provides a nice way to visualize data [39]. From simple pie charts to complex motion charts, the chart galley provides a large number of well-designed chart types.

Charts are exposed as JavaScript classes. Most charts are interactive and enable you to connect them to create a complex dashboard. Charts are rendered using HTML5/ Scalable Vector Graphics [42] technology to provide cross-browser compatibility and cross platform portability to handheld devices such as an iPhone, Android or tablet. SVG is a markup language for describing two-dimensional graphics applications and images, and a set of related graphics script interfaces.

5.2.1.2.1 Motion Chart

Motion Chart is a dynamic Flash chart used to show a series of data over time [43]. In this MQP, the Motion Chart was used to show the statistics of EMC's storage system capacity over a period of time. Figure 5-12 shows an example of motion chart.

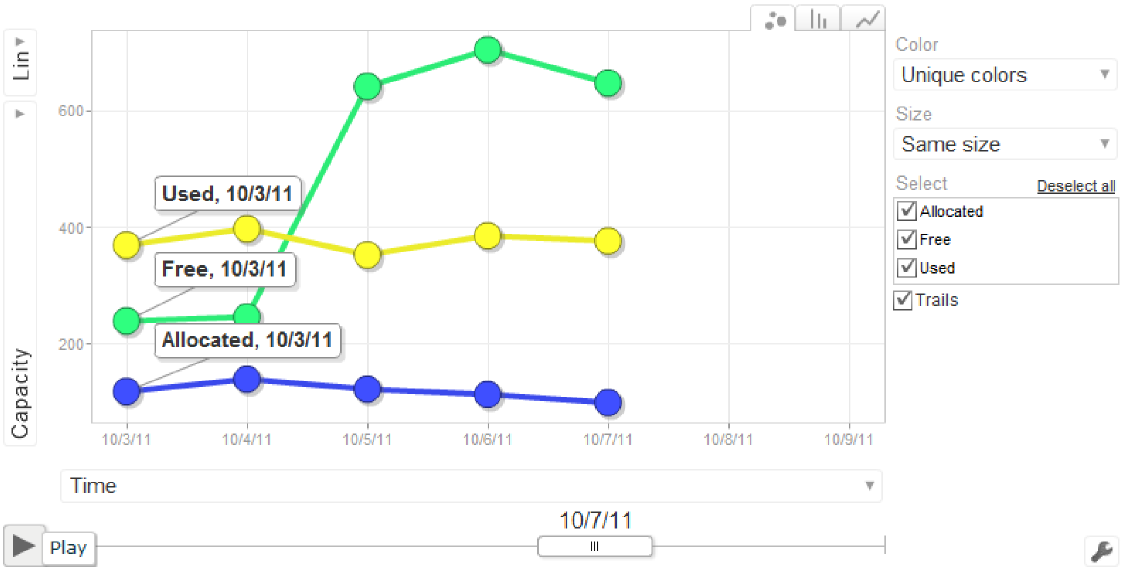
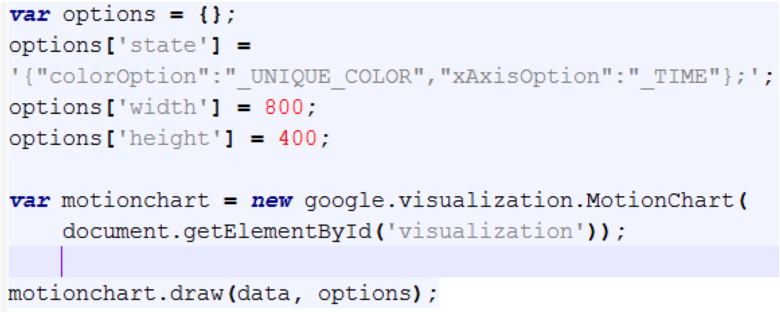


Figure 5-12 Using Motion Chart to Display Storage Stats over Time [35]

It is noted that the motion chart is very flexible as it can be configured by JavaScript. For instance, changing the color of each node and turning on/off the trails were not hard to achieve. Figure 5-13 gives an example of the code for drawing a motion chart using Google's library.

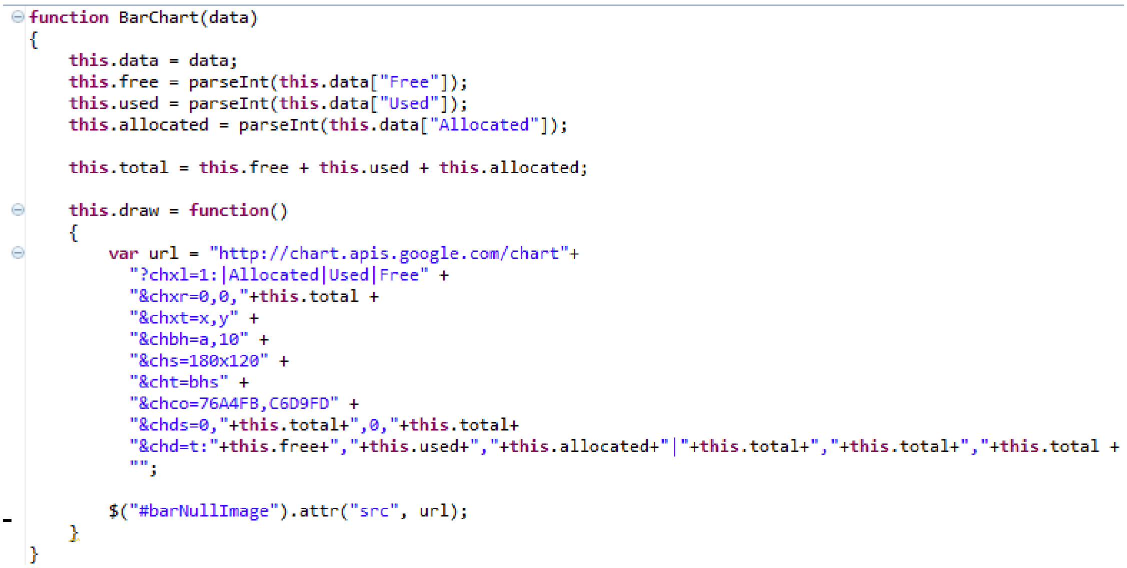


```
var options = {};
options['state'] =
'{"colorOption":"_UNIQUE_COLOR","xAxisOption":"_TIME"};';
options['width'] = 800;
options['height'] = 400;

var motionchart = new google.visualization.MotionChart(
    document.getElementById('visualization'));

motionchart.draw(data, options);
```

Figure 5-13 Code snippets of drawing a motion chart [43]

5.2.1.2.2 Bar Graph & Sparkline

A bar chart in the left panel shows the consumption of the storage system [44]. It basically gives users a sense of how full a particular storage is. Bar Graph is an image chart API provided by Google. As indicated in the source code in Figure 5-14, the data was gathered and eventually formed into a Uniform Resource Locator (URL) address. The Google API render the image based on the data provided and returns an image. Then, the program changes an image tag using the jQuery library.



```
function BarChart(data)
{
    this.data = data;
    this.free = parseInt(this.data["Free"]);
    this.used = parseInt(this.data["Used"]);
    this.allocated = parseInt(this.data["Allocated"]);

    this.total = this.free + this.used + this.allocated;

    this.draw = function()
    {
        var url = "http://chart.apis.google.com/chart"+
          "?chxl=1:|Allocated|Used|Free" +
          "&chxr=0,0,"+this.total +
          "&chxt=x,y" +
          "&chbh=a,10" +
          "&chs=180x120" +
          "&cht=bhs" +
          "&chco=76A4FB,C6D9FD" +
          "&chds=0,"+this.total+",0,"+this.total+
          "&chd=t:"+this.free+","+this.used+","+this.allocated+"|"+this.total+","+this.total+","+this.total +
          "";

        $("#barNullImage").attr("src", url);
    }
}
```

Figure 5-14 BarChart.js – A JavaScript Object that handles getting a bar chart image from Google

Sparklines are simple lines that can represent the trend of data. They are drawn based on a series of one-dimensional data. For instance, if the data set includes “10, 20, 40, 70”, the sparkline will be a curve that curve up. Three sparklines were rendered on the left panel of the screen to show the recent trend of statistics of the storage systems. They were very good at giving the users a quick overview of the trend over time. Figure 5-15 shows how bar chart and sparklines are used in the program.

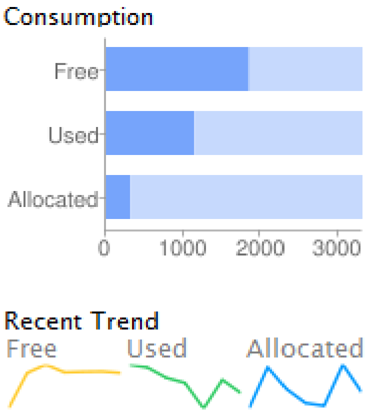
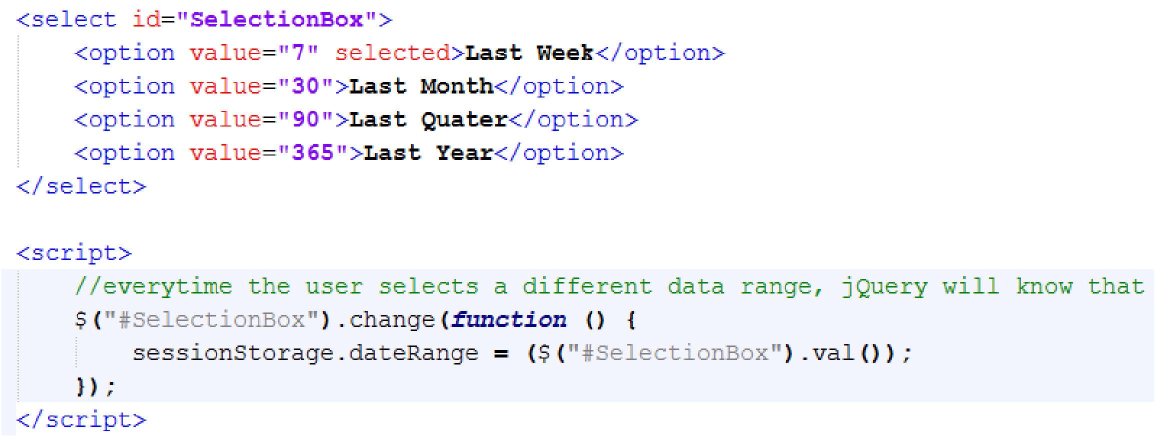


Figure 5-15 Bar Chart and Sparklines [35]

5.2.1.3 *JavaScript & jQuery Tools*

JavaScript was used almost everywhere in this part of the implementation [45]. JavaScript is the most commonly-used scripting language for the Web. It is front-end language that enables users to interact with webpages more easily. jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

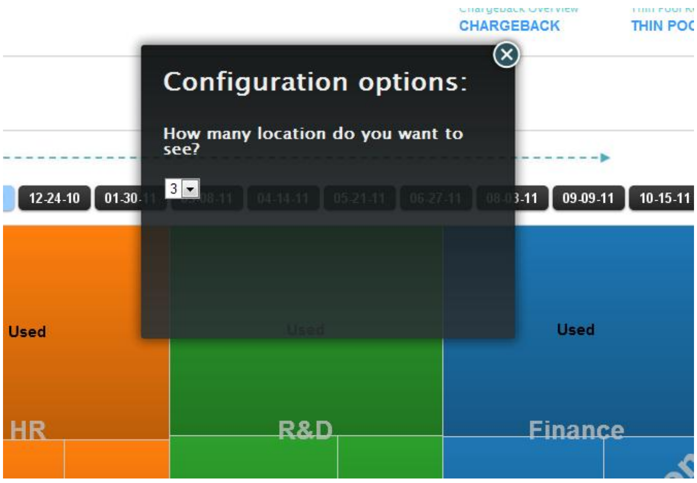


```
<select id="SelectionBox">
    <option value="7" selected>Last Week</option>
    <option value="30">Last Month</option>
    <option value="90">Last Quater</option>
    <option value="365">Last Year</option>
</select>

<script>
    //everytime the user selects a different data range, jQuery will know that
    $("#SelectionBox").change(function () {
        sessionStorage.dateRange = ($("#SelectionBox").val());
    });
</script>
```

Figure 5-16 Sample jQuery Usage

In the example in Figure 5-16, jQuery binds the dropdown selection box with an event. Every time the user selects a different date range, jQuery gets the new date range and stores it in Session Storage.

In addition, an Overlay API [46] from flowplayer.com was used. It is a jQuery tool that creates nice-looking and interactive webpage. The Overlay effect was used to display the configuration options for the visualizations. This way, users can customize this visualization tool more easily. Figure 5-17 shows an overlay effect.



Configuration options:

How many location do you want to see?

3

Figure 5-17 Overlay Effects [46]

5.3 Consumption Overview

In order to represent the consumption over time, the percentage and the actual amount of usage should both be displayed. Due to the different properties of these types of data, different representations were created. As mentioned in section 4.4.4, the consumption was presented with two graphs: a concentric ring graph and a line area graph. Both were designed by a team called Axiis which provides an open source package that includes the basic implementation of two graphs. This open source package was built based on Adobe Flex and Degrafa's Library.

5.3.1 Adobe Flex

Adobe Flex is a software development kit (SDK) for the development and deployment of cross-platform rich Internet applications based on Adobe Flash Platform. Flex provides the ability to do data visualization and UI interactions on the client-side, so it spends less time waiting for a server to respond. This offloading and client-side state handling makes back-ends more scalable and efficient [47].

In this project, the consumption overview graphs were written and modified in Adobe Flash Builder with Flex. Flex was used to combine the two graphs into a single file. A main file was created to hold the graphs and the link between them to perform the simultaneous highlight on both of them. The connection of two graphs is controlled by a mouse event listener, and it sends messages back and forth between the two graphs.

The Flex application was embedded in HTML as a “swfobject". Flex enables communication between ActionScript and the HTML page containing JavaScript and an embedded SWF file; this communication is called ExternalInterface. The communication is available in both directions. This allows the graphs built in Flex to be more interactive. In this project, ExternalInterface was used in both use case 2 & 3. The information shown on the left panel was pulled out from the SWF file with ExternalInterface [48]. This approach is straightforward and time sufficient. Instead of going back to the database to obtain the data, it can be received immediately. ExternalInterface was defined in Flex as follows:

```
ExternalInterface.call("displayMonth2", axiisSprite.data.month,
axiisSprite.data.name, nf.format(axiisSprite.data.share * 100) + "%");
```

`"displayMonth2"` was the name of the function defined in HTML as following:

```
<script>
    // call back functions
    function displayMonth2(s,d,r)
    {             document.getElementById('month').innerHTML = s;
        document.getElementById('status').innerHTML = d;
    document.getElementById('number').innerHTML = r;
    }
</script>
```

5.3.2 Axiis

Axiis[49] is an open source project focusing on data visualization. Both of the consumption graphs were originally from the Axiis project. It contains pre-built visualization components as well as abstract layout patterns and rendering classes that were useful in implementing this project. Axiis is built upon the Degrafa's library[50].

Figure 4-10 and Figure 4-11 were the original examples from Axiis. There were some unnecessary features that need to be removed. The trend example had a mouse over function that highlights the whole area of the same category, as shown in Figure 5-18. This was removed since it was irrelevant to the purpose of the consumption overview. The color scheme of this graph agrees with the concentric rings. A line was suggested to be added along with the tooltip on the mouse over. The built-in VRule in Flex was chosen. It had some effects to make the vertical line appear gradually instead of instantly. This line was implemented by setting the y-value to be a constant, and x-value set by the mouse event.

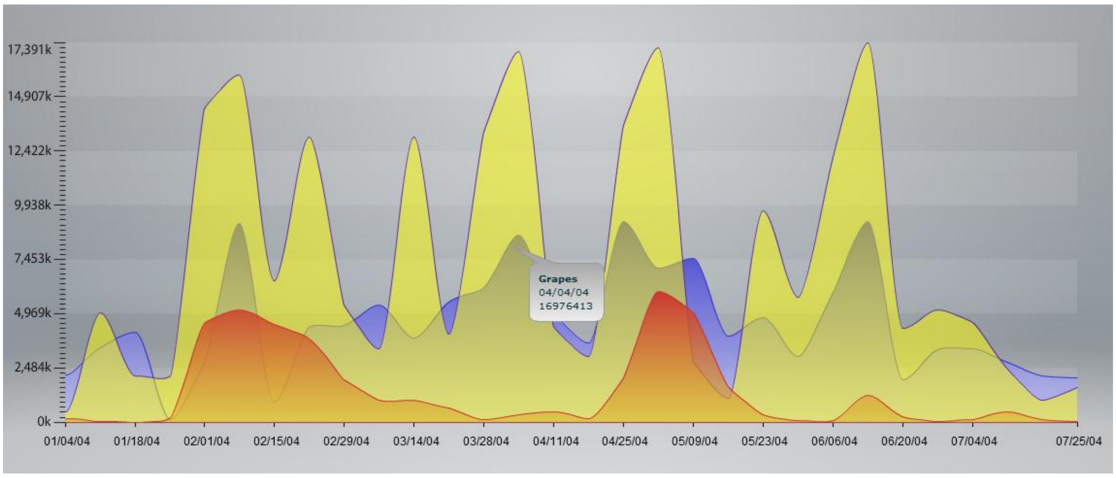


Figure 5-18 Basic Trend Graph Example with Highlighting [30]

5.3.3 Degrafa

Degrafa, Declarative Graphics Framework[50], is an open source project. The focus behind it is to bring the graphics classes up a level to provide a common ground between developer and designer within Flex, and enable the graphics classes to become first class citizens within the Flex framework. Degrafa can be used for Rich User Interfaces, Data Visualization, Mapping and Graphics Editing.

The grid of the concentric ring graph was implemented using Degrafa's geometry. The following example shows how to draw a vertical line with a blue color stroke on a black color gradient background, as shown in Figure 5-19.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application    xmlns:mx="http://www.adobe.com/2006/mxml"
                   layout="absolute"
                   backgroundGradientColors="[#333333, #222222]"
                   xmlns:degrafa="com.degrafa.*"
                   xmlns:paint="com.degrafa.paint.*"
                   xmlns:geometry="com.degrafa.geometry.*"
                   viewSourceURL="srcview/index.html">

    <degrafa:Surface     verticalCenter="0"
                         horizontalCenter="0">

        <!-- Creating Strokes.  -->
        <degrafa:strokes>
            <paint:SolidStroke    id="blue"
                                  color="#62ABCD"
                                  alpha="1"
                                  weight="3"/>
        </degrafa:strokes>

        <!-- Creating a Geometry Group.  -->
        <degrafa:GeometryGroup>

            <!-- Creating a Vertical Line.  -->
            <geometry:VerticalLine    stroke="{blue}"
                                      y="0"
                                      y1="200"/>

        </degrafa:GeometryGroup>
    </degrafa:Surface>
</mx:Application>
```

Figure 5-19 Example of a vertical line[51]

The grid's position and length were dependent on the data canvas that the concentric ring graph lies on to assure the grid would always lay on the ring graph correctly despite the size of the graph. With the initial implementation, the grid had a major defect: mousing-over the grid highlighted the last section of the outer ring with the tooltip. This was caused by drawing the grid on the wrong layer. The grid was moved to the ForegroundGeometric to fix this defect.

6 Evaluations

After implementation, the running prototype was presented to some subjects in order to conduct user studies. These studies were important to identify the pros and cons of the prototypes. The evaluation can be valuable to the EMC engineers because they can build better visualization tools based on the prototype and feedback from the subjects.

6.1 Study Subjects

The subjects were chosen by Dan Norton, the sponsor of this MQP. These subjects were software engineers from EMC Corporation. Some of them were Senior Software Engineers and some were from the User Experience Design (UXD) team. All had domain knowledge of EMC's products and the importance of these visualizations.

6.1.1 Study Protocol

- Each user study was conducted via a remote internet conference call. The MQP team sat in front of a computer at WPI, and the subjects were at EMC Corporation (at different offices). The entirety of the study was conducted over the internet because the subjects were very busy and some of them were physically located far away from the MQP team. Each study lasted around one hour.
- The MQP team told the subjects some background information about the project, such as the use cases.

- Three alternate designs for graphical visualizations of data were presented to the subjects. The MQP team shared their computer screen so that the subjects could see the team's computer screen. Short questions were asked in order to gather feedback and suggestions from the subjects.
- Subjects were encouraged to ask the researchers for more detailed information regarding the graphical visualizations, such as asking the purpose or how this might be implemented.

The detailed study protocol can be found in the Appendix.

6.2 Feedback from Subjects

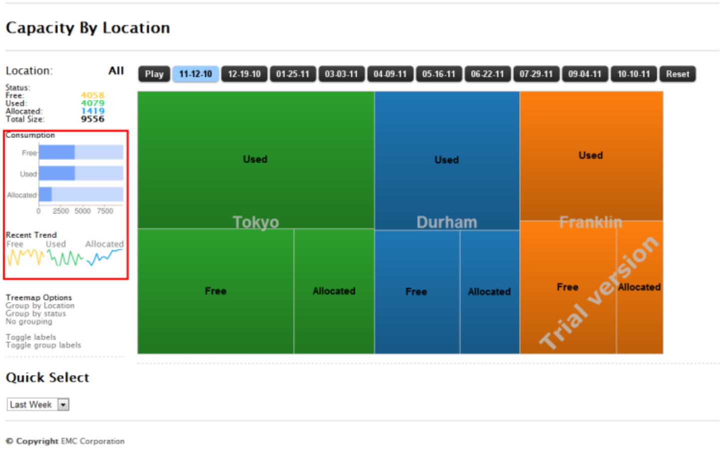
As mentioned earlier, five subject studies were conducted. It is important to note that in the study, a lot of comments, both positive and negative, were given by the subjects. However, more negative feedback was reordered and summarized in this section because they could help improve the visualizations.

6.2.1 Subject 1

The subject is a software engineer who has done extensive User Interface (UI) design. She provided some valuable feedback on the proposed visualization designs.

The subject thought the chargeback visualization is not very easy to understand. One of the reasons was that there was not any label or legend shown on the visualization tool. All that was shown on the graph was some cylinders with different colors. No indication of the meaning of each part was shown on the screen. Also, the subject maintained that they the colors differed too dramatically. Instead, she suggested that the tool should use the smallest conceivable different colors. In addition, she argued that it would be good to add some mouse-over functions, such as display some detail information about the elements that the user was pointing at. On the other hand, the subject liked the feature that the proportion of different sections (Primary LUNs, Local LUNs, and Remote LUNs) could change based on users' focus.

When the subject looked at the usage over time visualization, she had a hard time understanding the concentric ring graph on the left side. She maintained that she had an impression that the elements in the inner rings looked always smaller than the outer ones, despite the fact that the elements on the inner ring may represent a bigger percentage. However, she found the visualization on the right side, the trend graph, much more useful. She felt that the two graphs were trying to show the same data, but the trend graph was clearer and could give more information. Overall, she liked the color design and she felt that both of the visualization graphs could be very useful with some improvements.



Capacity By Location

Location: All

Status:
Free: 4058
Used: 4079
Allocated: 1419
Total Size: 9556

Quick Select: Last Week

Figure 6-1 Added titles for the two charts after Subject 1 pointed out that they were not clear without titles [35]

The subject generally liked the capacity used breakdown visualization because she felt she could get a lot of information from it. However, she was a little confused with the layout. The trend graph and bar graph on the left panel did not have any title, so the subject did not know what the data was about. Figure 6-1 shows the changes made based on this subject's input. Plus, she felt that there were too many colors on the page. On the other hand, she liked the motion chart and thought it was very interactive.

6.2.2 Subject 2

Subject 2 is a senior software engineer at EMC. He was very familiar with all the use cases and the related technologies. Based on his knowledge, he was able to give some very useful suggestions.

The layout of the Chargeback design was clear enough for the subject to understand the relationship between the primary, local and remote replicas. The animation of the cylinders moving was not well-defined and it caused confusion for the subject. The subject didn't understand the navigation of the chargeback visualization. He also suggested that the current layout would cause issues when there were too many elements in the visualization. It was mentioned by the subject that this design could be used for the FAST technology.

For the usage over time visualization, the subject commented that the visualization on the left was not as clear as the one on the right. The subject was able to read the left graph with the labels of the percentage, and then he agreed that the concentric ring was a better representation for percentage. Compared to the trend line area graph, the subject found the concentric ring was more interesting. The mouse-over function was not able to provide enough information according to the subject's response. He suggested that the mouse-over of the concentric ring should highlight the whole ring instead of a segment of it, so it could provide all the information of that particular period of time. The same suggestion was given to the trend line area graph; the subject thought it would be better to see all three (Used, Allocated and Free) instead of only one of them. In this case, these numbers could be compared to each other, and it would be more informative.

The third visualization, the bar chart on the left panel was found useful by the subject. The colors of the three graphs (bar chart, recent trend, and treemap) on this page were mixed and it would cause confusion. The subject did find the animation of the graph change interesting, and he suggested that there would be a better way to achieve it. When the graph was changed to Usage by Status, he suggested changing the label of each small section to its location instead of displaying the status of each section.

Overall, the subject found that the visualizations were helpful to understand the information better.

6.2.3 Subject 3

Subject 3 is a software engineer at EMC. His task is to get in touch with clients and understand what is wrong with the current design, and why this issue has caused the user problems. The experience of working with customers is invaluable for this evaluation, as this product would, if developed to completion; need to meet customer satisfaction in order to be considered a successful design.

At first glance of the chargeback visualization, Subject 3 was impressed with the design. He repeatedly stated that it was a solid design, and that the opportunity for growth in the development of it as a tool was very desirable. The background "highlighting" of whichever object was currently selected helped him to understand the current focus of the visualization at any given point in time, and the emphasis on the relationships between the objects, as opposed to the current topology map, was well received. The stretching "illusion" that was developed to allow the user to switch from one view to the next without difficulty was also very well-received, as he said that it gave the user a more fluid feel for the design. Scalability was another factor which he paid close attention to, and he mentioned the multi-remote-LUN backup systems which EMC often performs that would need to be accounted for.

When informed that this could be alleviated by simply adding another section, he was further impressed, and suggested future upgrades to traversing the visualization to the left and the right. He did mention that he wanted context sensitive functionality (mouse-over, right click, and other options which detailed information the user might need), as the complexity of the systems required consistently changing values to be readily accessible, and mentioned that he would like to see a more developed, production-level design that would demonstrate this visualization in action. In following up with that design, the subject also approved of the array design, although he said that he would like more details before he could make an informed decision.

The usage over time visualization was hesitantly received at first, as he saw the two graphs giving different perspectives of the same dataset initially was confusing. However, after the brief disorientation while working with the design, he found that the two graphs were actually very helpful in demonstrating the sum of all the information that he needed to have. The difference between the two graphs was very apparent to him, and unlike the other subjects, he appeared to have no pre-conceived notions about the increasing circumferences of the rings in the circular visualization being representative of ever increasing data. However, he did mention that the wild fluctuations that were present in the dataset due to the random generation of the visualization hampered his ability to understand what was being conveyed. Furthermore, he mentioned that the visual distinctions between the two graphs, as well as the line in the second graph, helped to get the point across to him better.

In the final visualization, of usage by item, the subject had significant difficulty in interpreting the system. The treemap design was readily apparent, but the scalability aspect of the design troubled him. He brought up several pieces of data which he believed should be represented by the visualization which were not readily accessible, which included the handling of RAW data versus usable data, a representation of fragmented data to give an accurate view of the spacing, as opposed to the simplistic three-square design layout that was available in this visualization at the time. Malleable materials, or materials whose values can change depending on certain requests, were also an issue. However, overall, while he did not overall prefer the design, he did not have too much difficulty in understanding it, and did mention that the design was decent for visualizing small datasets.

Subject 3 took considerable time to analyze each of the visualizations and get a solid view on what they represented. He was very thorough in his analysis, and asked questions when he was confused. This led to a large amount of data on the design gathered due to his help.

6.2.4 Subject 4

Subject 4 is a software engineer from EMC Corporation. He generally found the Chargeback visualization pretty good but he also gave some potential improvements. He pointed out that this visualization was only useful for storage administrators, but not any other users such as end users or managers. He was confused with some parts of the graphical representation because a few features were not working yet. He maintained that he had to really use the program and test the features to see if it was useful or not. The subject suggested that a similar visualization design could be very useful for displaying some FAST relationships. He showed us a graph (see Figure 6-2) that is similar to this visualization and he found that pretty useful.

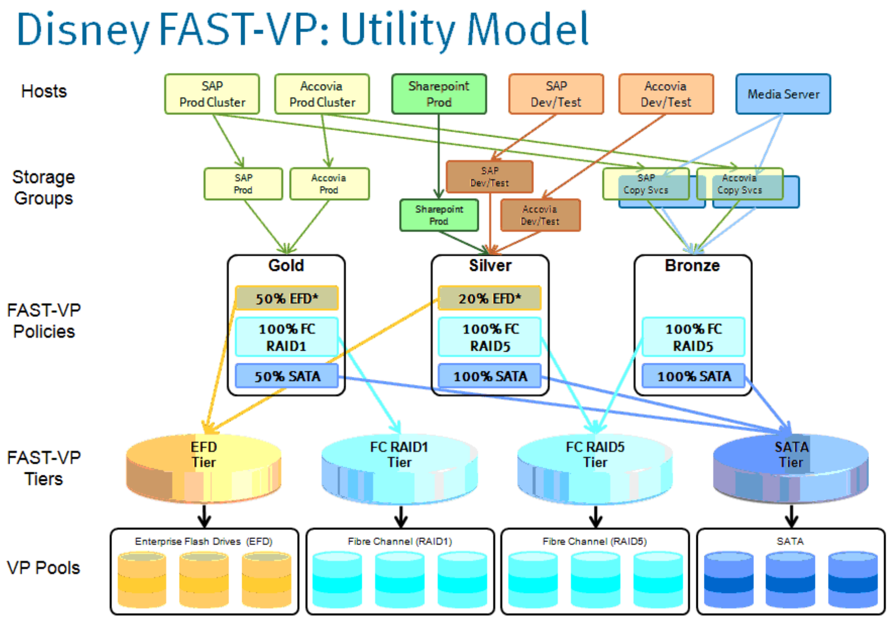


Figure 6-2 Similar design that Subject #4 found useful[52]

The subject suggested that using a classical area chart to visualize usage over time was clear and nice. However, he had a hard time understanding the concentric circle visualization on the left side. One reason was that he could not figure out what each ring meant in the concentric circle. The subject felt that reading from inner to outer rings was not the way his brain would work. He pointed out that his initial thought was that the center part of the concentric circle was more important so that the most recent data should be on the most inner ring.

The subject had never seen a treemap before, but he maintained that he could easily figure out what the treemap was trying to represent. He liked the color scheme. However, he felt that there could still be improvements. For instance, the treemap only showed each element’s value relative to other elements. It was good to compare which one is big. But he pointed out that customers might be more interested in the change in absolute size. Therefore, the ability of changing the total size of the treemap would be helpful.

6.2.5 Subject 5

Subject 5 is an engineer currently working on the EMC's User Experience Design team. He was able to give a number of good suggestions related to the User Interface of the prototype.

The subject was confused by the Chargeback visualization, and he was having problems finding the connection between cylinders in different color areas. He suggested that there was no dependence from orange to purple, but there was a relationship between green and orange, and green and purple, as shown in Figure 6-3. The single cylinder’s color was not clear, so he suggested adding a legend for different colors. The fourth color bar could be added as the subject suggested, and it could be used to group the applications. This designed was not scalable. The visualization was not easy for the user to understand each attribute in the supply chain. The relationship between primary, local and remote replicas would be clearer represented with a block design.

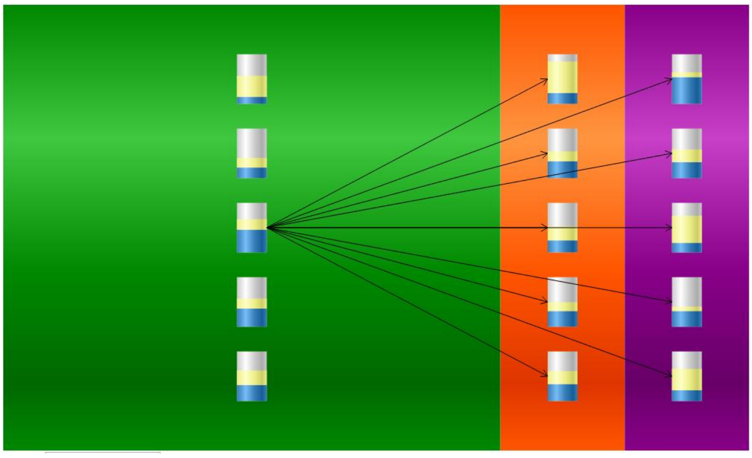


Figure 6-3 Three different colors, no dependence between orange and purple. [35]

For the second visualization, concentric rings and trend graph, the subject was not able to tell the advantage of the rings compared to the traditional trend graph. He suggested that a line graph that had the y-axis as percentage would be much more helpful than the concentric ring. The subject described the concentric rings as an unusual way of representation and didn't add much overall. The rings didn't have time labels but only the start and the end time labels as shown in Figure 6-4. Having two similar charts overlay with separate axes would be a better representation than the current design.

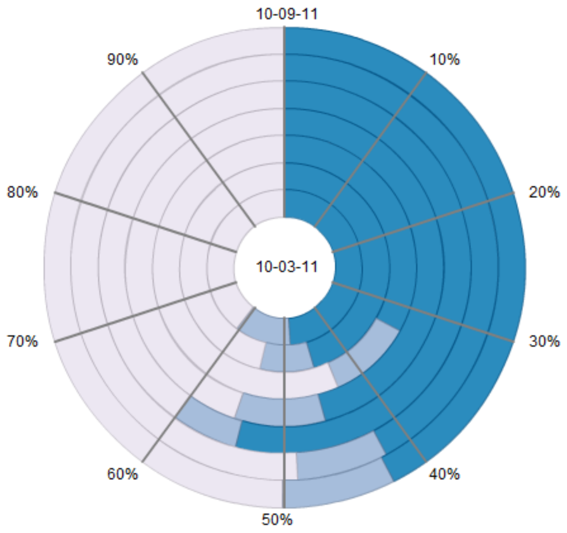


Figure 6-4 Concentric rings with only two time labels [35]

The subject seemed to like the Usage by Value visualization. The recent trend graph on the left panel was said to be very useful. He was uncertain about the play functionality on the treemap; during play, it was hard to keep track of the dynamic changes of each small section. The subject thought the color scheme of the visualization should consider the potential color blind users. A suggestion of having a relative comparison between trends over time for different locations was given by the subject.

Overall, the subject suggested that the prototype should provide a way to show the relevant data without a lot of clicks.

6.3 Summary of Evaluation

Overall, there was a great deal of differences in opinions on the design. While there will not be a consensus on the design, there was a fair amount of differentiation between views from one user to the next. The User Interface subjects focused strongly on colors, and requested additional or muted color schemes, while other subjects preferred the color ranges as it made the distinctions between groups easier to view. Others preferred a layout scheme, such as that of the chargeback, over the other visualizations. However, the designs overall, though critiqued and informed that there was room for improvement, were favorably received. Each subject had improvements which they offered, but none of the designs were considered to be a “bad” or “incorrect” implementation of the datasets given. In fact, a common word used to describe each of the designs was “interesting,” which could be interpreted either way. The users did note their appreciation of the visualizations overall.

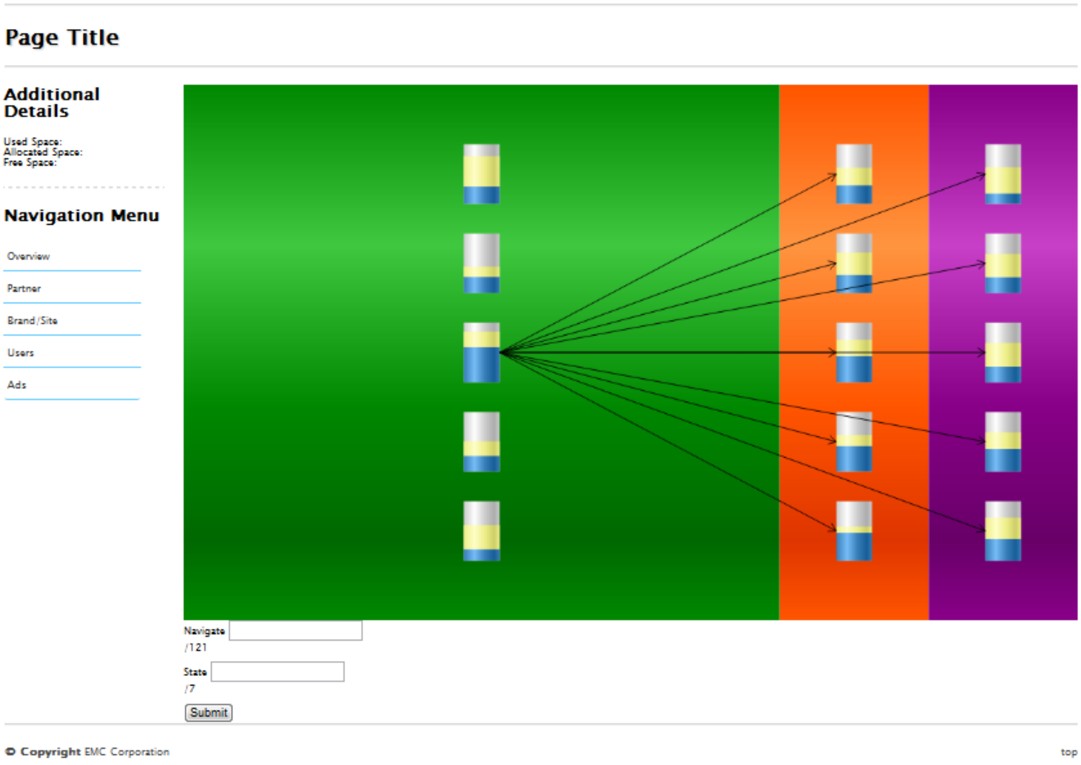
7 Conclusion

The MQP team accomplished the project over approximately 8 weeks. The main objective of this project was to come up with useful and effective visualization ideas for EMC's networked storage systems. Overall, the MQP team managed to accomplish most of the tasks, but there is some future work that can be done to improve the project even more.

7.1 Major Accomplishment

As a result, the MQP team made themselves familiar with the EMC's technology. They also had a deeper understanding of visualization data techniques, including various ways of representing complex data.

In addition, the MQP team came up with various ways of visualizing the storage systems based on the given use cases. A final design was achieved after a few iterations. Some suggestions from EMC employees and professors were taken in order to improve the design. A prototype was implemented in order to show the visualization ideas in a better way. In this process, the team members improved themselves by learning new technology and programming languages, such as Flex [53], PHP[40] and HTML5[32]. Figure 7-1, Figure 7-2, Figure 7-3 are the sample screenshots of the prototype[35].



Page Title

Additional Details

Used Space:
Allocated Space:
Free Space:

Navigation Menu

- Overview
- Partner
- Brand/Site
- Users
- Ads

Navigate /121

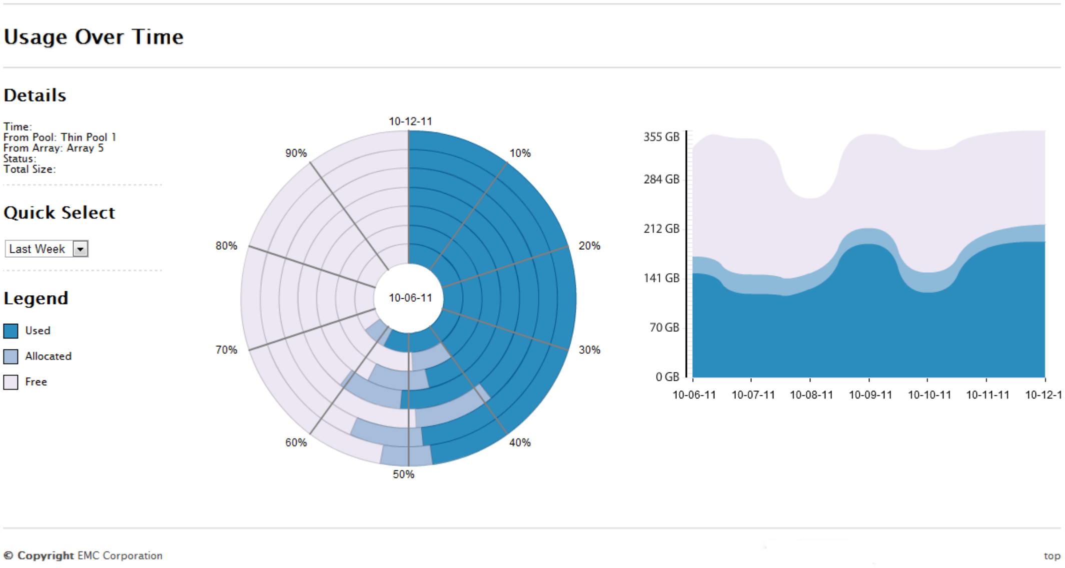
State /7

Submit

© Copyright EMC Corporation

top

Figure 7-1 Chargeback Visualization [35]



Usage Over Time

Details

Time:
From Pool: Thin Pool 1
From Array: Array 5
Status:
Total Size:

Quick Select

Last Week ▾

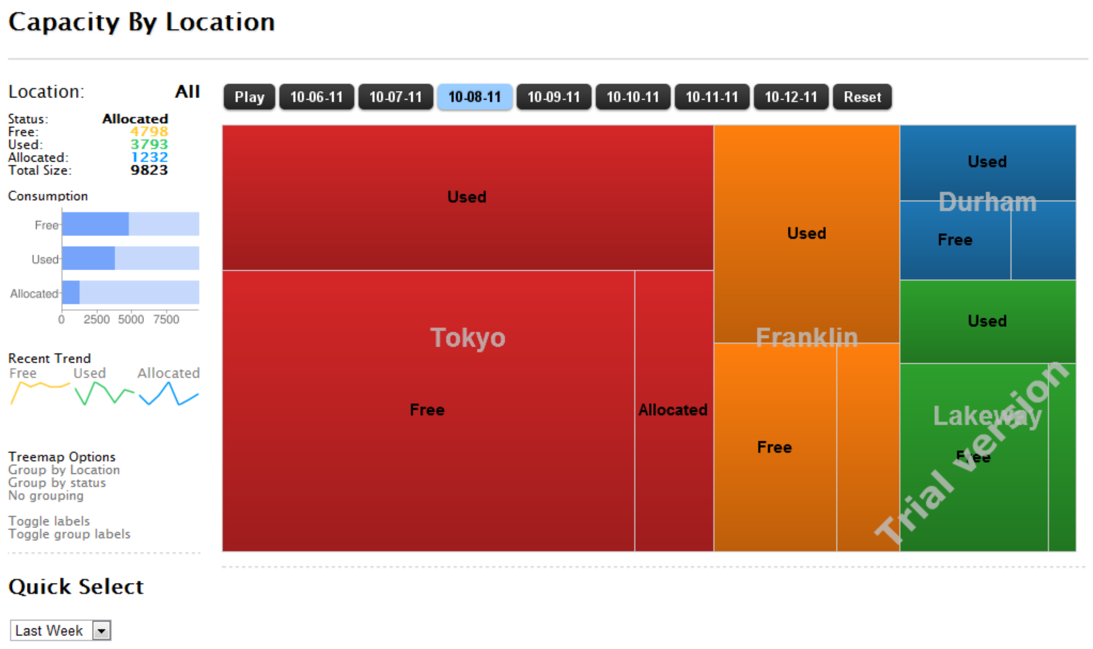
Legend

- 🟦 Used
- ▨ Allocated
- □ Free

© Copyright EMC Corporation

top

Figure 7-2 Used over Time Visualization [35]



Capacity By Location

Location: All

Status: Allocated
Free: 4798
Used: 3793
Allocated: 1232
Total Size: 9823

Figure 7-3 Capacity Used by Value Visualization [35]

User studies were conducted in order to obtain more feedback. In total, five user studies were conducted and the feedback was recorded and summarized into this report. In addition, some suggestions obtained from the user studies were adopted. Eventually, a final presentation on this MQP was given at EMC Corporation.

7.2 Evaluation Summary

Feedback, both positive and negative, was given regarding the design and implementation of the visualization ideas. Generally, people understood and liked the design ideas. They thought these kinds of data visualization were useful to solve the use cases. Meanwhile, they all provided suggestions on some details or part of the prototype. Those suggestions and feedback were very important and could definitely improve the visualization to some extent. Detailed evaluation can be found in Section 6.

7.3 Future Work

This section describes some possible work that could be done to expand upon this project.

7.3.1 Modifications to Design

There are a lot of ways to modify the design for future projects. The Chargeback visualization needs to be more informative. For examples, all of the three colors were not defined anywhere for the Chargeback visualization. One way to solve this is put the labels for each section on the top of the color areas. The way of representing the relationship between Primary, Local and Remote Replica is not sufficiently. The relationship between Primary and Local, or Primary and Remote is very clear defined by the arrows. In theory, the Local and Remote Replica don't have relationship. This should be clear in the future design.

The Consumption over Time graph can be represented by a line graph overlay a line area graph that represents the percentage. The line graph will replace the concentric rings. The concentric ring may be a better way to visualize the percentage, but it takes time for the users to understand. The line graph is not ideal yet, as the total amount of storage increases, the line graph may confuse the users. With the storage increases, the percentage will be based on different total amount of storages. The Used percentage line is not corresponding to the actual amount of storages, and then the percentage line graph doesn't mean anything.

Although they are overlain on each other, they are still two different graphs with no relationships. Use one graph represents two sets of data will result a lot of lines and colors in one graph. This might be overwhelming to the users.

The Usage by Value visualization can be improved by adding a drill-down function. The drill-down will bring up another layer of the data to provide detailed information about each section. The visualization would then be more informative. The color of the treemap and other charts on this page should be corresponded to each other.

7.3.2 More Use Cases

Only three use cases were chosen for this project. Future projects should consider additional or different use cases, in order to give EMC additional details and more ideas for how this visualization will help them interpret the information that is contained in their visualizations better.

7.3.3 User Study

The best way to get users' feedback is to let them use the prototype demo by themselves, in conjunction with their hardware. Feedback from different people will be really helpful to improve the design as well. The programmers should not be the sole demo-ers, however, as they are not the final users for the product, and should not be treated as such. The effectiveness of the design is decided by users, and some possible user studies can be done in future projects. Prospective users should be chosen as the targets before the testing commences, however.

7.3.4 Use Real Data

It’s important to visualize representative data. During this project, the majority of the data that was used in the demos were randomly generated; these random values had the unintended side effect of confusing the users. Real data may help the average tester to understand the visualization without being stopped by the information available, so real data should be used in future projects. The users should also be familiar with the data, and understand how the current systems look like, and how the new demo should look, too. If the graph is too different from what they expect, they may need to focus on the fake data instead of the visualization. Understanding when each separate dataset is necessary is critical to have the real data for the user studies.

Bibliography

[1]. **Komiega, Kevin.** EMC Banks Big Storage Customers. [Online] SearchStorage, Dec 17, 2002. [Cited: Sep 23, 2011.] http://searchstorage.techtarget.com/news/869712/EMC-banks-big-storage-customers.

[2]. IT's new holy grail: Break out of the 70% maintenance loop. *TechRepublic.* [Online] May 11, 2010. [Cited: September 19, 2011.] http://www.techrepublic.com/blog/hiner/its-new-holy-grail-break-out-of-the-70-maintenance-loop/4283.

[3]. *Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information Structures.* **Johnson, Brian and Shneiderman, Ben.** IEEE Computer Society Press : San Diego, 1991. IEEE Visualization Conference. pp. 284-291.

[4]. **Ward, Matthew O., Grinstein, Georges and Keim, Daniel.** *Interactive Data Visualization: Foundations, Techniques, and Applications.* A K Peters Ltd : Natick, 2010.

[5]. *Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations.* **Stasko, John and Zhang, Eugene.** 2000, Information Visualization, IEEE Symposium, p. 57.

[6]. *Tidy Drawings of Trees.* **Wetherell, Charles and Shannon, Alfred.** 5, September 1979, IEEE Transactions on Software Engineering, Vol. 5.

[7]. **Robertson, George G., Card, Stuart K. and Mackinlay, Jack D.** Information Visuazliation Using 3D Interactive Animation. *Communications of the ACM.* April 1993, Vol. 36, 4.

[8]. *The Hyperbolic Browser: A Focus+Comtext Technique for Visualization Large Hierarchies.* **Lamping, John and Rao, Ramana.** 1996, Journal of Visual Languages and Computing, pp. 33-55.

[9]. *Cone Trees: animated 3D visualizations of hierarchical information.* **Robertson, George G., Mackinlay, Jock D. and Card, Stuart K.** ACM : New York, 1991, pp. 189-194. 0-89791-383-3.

[10]. **Telea, Alexandru C.** *Data Visualization - Principles and Practice.* A K Peters. Ltd : Wellesley, 2008. 978-1-56881-306-6.

[11]. **Harris, Robert L.** *Information Graphics - A Comprehensive Illustrated Reference.* Oxford University Press : Atlanta, 1999. 0-19-513532-6.

[12]. About EMC. [Online] EMC Corporation. [Cited: September 20, 2011.] http://www.emc.com/about/.

[13]. Storage Management and Information Management Products. [Online] [Cited: September 20, 2011.] http://www.emc.com/products/category/subcategory/storage-resource-management.htm.

[14]. **Norton, Dan.** *WPI EMC Project Kickoff SCI.* Hopkinton : EMC Corporation, 2011. Microsoft Powerpoint.

[15]. **Tate, Jon, Lucchese, Fabiano and Moore, Richard.** *Introduction to Storage Area Networks.* 4th Edition. s.l. : International Business Machines Corporation, 2006. p. 2.

[16]. *Engineering White Paper - Storage Architecture Terminology.* Hopkinton : EMC Corporation, 2004. White Paper.

[17]. *Engineering White Paper - Virtual Provisioning Terminology.* Hopkinton : EMC Corporation, 2008. White Paper.

[18]. *Engineering White Paper - Understanding the Differences between Accessible and Allocated Metrics.* Hopkinton : EMC Corporation, 2004. White Paper.

[19]. *EMC Clariion CX500 Networked Storage System.* Hopkinton : EMC Corporation, 2005. Specification Sheet.

[20]. Computer and Communications Experts. *America's IT Pros.* [Online] [Cited: October 1, 2011.] http://www.americasitpros.com/.

[21]. *Engineering White Paper – EMC ControlCenter 5.2 Storage Allocation Terminology.* Hopkinton : EMC Corporation, 2004. White Paper.

[22]. About Symantec. [Online] Symantec. [Cited: September 20, 2011.] http://www.symantec.com/about/.

[23]. **Gonzalez, Tom.** Horizontal Chart - Cluster with Single Column. *Axiis.* [Online] [Cited: October 1, 2011.] http://axiis.org/examples/HClusterColumnExample.html.

[24]. **Smith, David.** How pie charts fail. *Revolutions.* [Online] August 26, 2009. [Cited: October 1, 2011.] http://blog.revolutionanalytics.com/2009/08/how-pie-charts-fail.html.

[25]. **Ross, Greg.** Visualization: a Magic Table. *Google Chart API.* [Online] [Cited: October 2, 2011.] http://magic-table.googlecode.com/svn/trunk/magic-table/google_visualisation/example_1.html.

[26]. **Google.** Google Maps. [Online] Google Inc. [Cited: October 1, 2011.] http://maps.google.com.

[27]. **Du, Xiao, Liu, Zhaochen and Rolle, David.** Prototype Design PowerPoint. October : 1, 2011.

[28]. **Nielsen, Jakob.** F-Shaped Pattern for Reading Web Content. *useit.com.* [Online] April 17, 2006. [Cited: September 2011, 2011.] http://www.useit.com/alertbox/reading_pattern.html.

[29]. Visualization: DrasticTreemap. *Google Chart API.* [Online] Drastic Treemap. [Cited: October 1, 2011.] http://www.drasticdata.nl/DrasticTreemapGApi/index.html.

[30]. **Gonzalez, Tom.** Horizontal Chart - Line and Area. *Axiis.* [Online] [Cited: October 1, 2011.] http://axiis.org/examples/LineAreaSeriesExample.html.

[31]. **VanDaniker, Michael.** W3schools.com's Historical Browser Statistics - Custom Visualization. *Axiis.* [Online] [Cited: October 1, 2011.] http://axiis.org/examples/BrowserMarketShare.html.

[32]. HTML5 - A vocabulary and associated APIs for HTML and XHTML. *World Wide Web Consortium.* [Online] [Cited: October 5, 2011.] http://dev.w3.org/html5/spec/Overview.html.

[33]. **whatwg.org.** HTML Living Standard:. [Online] 10 18, 2011. [Cited: 9 15, 2011.] www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html.

[34]. DJ-NOTE. *HTML 5 Canvas Linear Gradient Tutorial [Basic].* [Online] January 21, 2011. [Cited: October 3, 2011.] http://dj-note.blogspot.com/2011/01/html-5-canvas-linear-gradient-tutorial.html.

[35]. **Du, Xiao, Liu, Zhaochen and Rolle, David.** *Visualized Networked Storage.* [Online] October 12, 2011. [Cited: October 15, 2011.] http://www.url66.com/emcmqp.

[36]. **Sarris, Simon.** Making and Moving Selectable Shapes on an HTML5 Canvas: A Simple Example. [Online] 08 23, 2010. [Cited: 09 15, 2011.] http://simonsarris.com/blog/140-canvas-moving-selectable-shapes.

[37]. How to give cone or cylinder a 3d effect using javascript on the canvas? And how to fill the desired color in them? *StackOverflow.* [Online] 3 26, 2011. [Cited: 9 18, 2011.] http://stackoverflow.com/questions/5362047/how-to-give-cone-or-cylinder-a-3d-effect-using-javascript-on-the-canvas-and-how.

[38]. **DrasticData.** Documentation Drastic Treemap. *Drastic Data.* [Online] 1.2. [Cited: October 2, 2011.] http://www.drasticdata.nl/DrasticTreemap/Documentation.htm.

[39]. Introduction to Using Chart Tools. *Google Code.* [Online] Google. [Cited: October 5, 2011.] http://code.google.com/apis/chart/interactive/docs/index.html.

[40]. PHP Tutorial. *W3schools.* [Online] [Cited: October 1, 2011.] http://www.w3schools.com/php/.

[41]. SWFObject Project. *Google Code.* [Online] Google. [Cited: October 5, 2011.] http://code.google.com/p/swfobject/.

[42]. Scalable Vector Graphics. *World Wide Web Consortium (W3C).* [Online] [Cited: October 8, 2011.] http://www.w3.org/Graphics/SVG/.

[43]. Visulization: Motion Chart. *Google Code.* [Online] Google. [Cited: October 8, 2011.] http://code.google.com/apis/chart/interactive/docs/gallery/motionchart.html.

[44]. Visualization: Sparkline (Image). *Google Code.* [Online] Google. [Cited: October 8, 2011.] http://code.google.com/apis/chart/interactive/docs/gallery/imagesparkline.html.

[45]. JavaScript Tutorial. *w3schools.com.* [Online] [Cited: October 8, 2011.] http://www.w3schools.com/js/.

[46]. Minimal setup for overlay. *jQuery Tools.* [Online] Flowplayer. [Cited: October 6, 2011.] http://flowplayer.org/tools/demos/overlay/index.html.

[47]. "What is Flex". *Adobe Flex Developer Resource.* [Online] [Cited: October 1, 2011.] http://flex.org/what-is-flex/.

[48]. External Interface. *Adobe Learning Center.* [Online] [Cited: October 7, 2011.] http://learn.adobe.com/wiki/display/Flex/External+Interface.

[49]. **Axiis.** Axiis. *Open source Data Visualization.* [Online] [Cited: October 1, 2011.] http://www.axiis.org.

[50]. Degrafa. *About Degrafa.* [Online] [Cited: October 2011, 9.] http://degrafa.org/about/.

[51]. Vertical Line Example. *Degrafa.* [Online] [Cited: October 2011, 9.] http://degrafa.org/source/HorizontalLine/VerticalLine.html.

[52]. **Anonymous.** *MQP User Study.* [interv.] Xiao Du, Zhaochen Liu and David Rolle. October 10, 2011.

[53]. Flex Overview. *Adobe Flex.* [Online] Adobe Corporation. [Cited: October 11, 2011.] http://www.adobe.com/products/flex.html.

[54]. **Eppler, Ralph Lengler & Martin J.** Periodic Table of Visualization Methods . *Visual-Literacy.org.* [Online] [Cited: 9 9, 2011.] http://www.visual-literacy.org/periodic_table/periodic_table.html#.

[55]. **Spence, Robert.** *Information Visualization.* Pearson Education Ltd : Broadway, 2011. (978) - 0201596261.

[56]. **Chen, Chaomei.** *Information Visualization and Virtual Environments.* Springer-Verlag : London, 1999. 1-85233-136-4.

[57]. **Soukup, Tom and Davidson, Ian.** *Visual Data Mining - Techniques and Tools for Data Visualization and Mining.* John Wiley & Sons, Inc : New York, 2002. 0-471-14999-3.

[58]. *PerfViz: A Visualization Tool for Analyzing, Exploring, and Comparing Storage Controller Performance Data.* **Sawant, Amit P., Vanninen, Matti and Healey, Christopher G.** International Society for Optical Engineering : New York, 2007, PROCEEDINGS- SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING, Vol. 6495.

Appendix A - Sample User Study Consent Form

Informed Consent Agreement for Participation in a Research Study

Investigator: Professors Matthew Ward and Emmanuel Agu, Students Xiao Du, David Rollé, Zhaochen Liu

Contact Information: xiaodu@wpi.edu, dsrolle@wpi.edu, jj.liu@wpi.edu

Title of Research Study: Prototype Visualization of Networked Storage

Sponsor: EMC^2

Introduction (recommended)

You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study: The purpose of this study is to develop a better understanding of what the users are looking for in a design focused on visualizing networked storage

Procedures to be followed: The subject will be required to observe and interact with the program that they are given. The time required for interaction will be between 20 and 40 minutes.

Risks to study participants: While no risks are anticipated, there may be unforeseen side-effects due to exposures to certain colors or animations. If you suffer from epilepsy or color-blindness, there is a possibility that viewing this program will adversely affect you.

Benefits to research participants and others: There are no benefits to you, the subject, but this study will help future forays into visualizations of networked storage.

Record keeping and confidentiality: (There will not be any recording, either audio or visual, that will be made. All data obtained will be done through written means, transcribed by the student investigators. These records will include names, but any personal information will not be referenced either in the paper or the final product. The only people with access to personal information will be the primary and co-investigators, student investigators, and EMC personnel. Confidentiality of the details will not be necessary, and details recorded will only be observations based on user's preference of the interface provided. Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury: You do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact:

Student Investigator: Xiao Du, Tel. 508-762-6965, Email: xiaodu@wpi.edu

Student Investigator: David Rollé, Tel. 303-250-4474, Email: dsrolle@wpi.edu

Student Investigator: Zhaochen Liu, Tel. 617-297-7799, Email: jj.liu@wpi.edu

Principal Investigator: Professor Matthew Ward, Tel. 508-831-5671, Email: matt@cs.wpi.edu

Co-Investigator: Professor Emmanuel Agu, Tel. 508-831-5568, Email: emmanuel@cs.wpi.edu

IRB Chair: Professor Kent Rissmiller, Tel. 508-831-5019, Email: kjr@wpi.edu

University Compliance Officer: Michael J. Curley, Tel. 508-831-6919, Email: mjcurley@wpi.edu

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

______________________________ Date: ____________________

Study Participant Signature

____

Study Participant Name (Please print)

____ Date: ____________________

Signature of Person who explained this study

Additional clauses to add to Consent Agreements, as appropriate:

The treatment or procedures used in this research may involve risks to the subject (or to an embryo or fetus, if the subject is or may become pregnant), which are currently unknown or unforeseeable.

Additional costs to the subject that may result from participation in this research include: (list).

Significant new findings or information, developed during the course of the research, may alter the subject's willingness to participate in the study. Any such findings will be promptly communicated to all research participants.

Should a participant wish to withdraw from the study after it has begun, the following procedures should be followed: (list). The consequences for early withdrawal for the subject and the research are: (list).

Special Exceptions: Under certain circumstances**,** an IRB may approve a consent procedure which differs from some of the elements of informed consent set forth above. Before doing so, however, the IRB must make findings regarding the research justification for different procedures (i.e. a waiver of some of the informed consent requirements must be necessary for the research is to be "practicably carried out.") The IRB must also find that the research involves "no more than minimal risk to the subjects." Other requirements are found at 45 C.F.R. §46.116.

Appendix B - User Study Protocol

In each user study, the following procedure will be used:

A. Each user study will be conducted via a remote internet conference call, and, during the conversation, the sponsor, Dan Norton, EMC liaison to WPI, will inform the subjects of their rights under the IRB's forms. The student researcher will sit in front of a computer at WPI, and the subjects will either be at EMC Corporation in Hopkinton (176 South Street, Hopkinton, MA 01748), or at one of several of EMC's project sites throughout the world, in a conference room requested by Dan Norton. The entirety of the study will be conducted over the internet, due to the distance some of the subjects live from the WPI students.

B. Three alternate designs for graphical visualizations of data will be presented to the subjects in the form of an interactive web page. The researchers will share their computer screen so that the subjects can see the researchers' computer screen. The subjects may use their mouse to control the screen (e.g. click the buttons on the interface)

C. Each user study will take less than 60 minutes of participation, and all data which is obtained from the subjects will only be written down in an informal, qualitative format so as to keep incidental or personal information secret.

D. The researchers will talk to five subjects under the supervision of the MQP's head sponsor, Dan Norton, a manager from EMC Corporation.

E. Some instructions of how to use this visualization tool will be provided as the user asks what a function is or how to perform a certain task. Each of these requests will be recorded and interpreted by the researchers at a later time.

F. A group of questions will be used in order to get feedback and suggestions from the subjects after the user study has been completed. They will also be informed that the full waiver will be sent to them shortly. These questions will pertain to the design of the visualizations, such as color schema, layout, scalability, and overarching UI design, and will not contain any information relating to the user's tasks or their jobs at EMC. Examples of these questions include: "Is the layout pleasing?", "Is it helpful to see the snapshot of the available arrays?", and others. Questions will not be strictly adhered to, as they are guidelines for future inquiry, and the actual responses will not be included in the report.

G. Subjects will be encouraged to ask the researchers for more detailed information regarding the graphical visualizations, such as asking purpose or how this might be implemented, by informing them of information if they have proven (via 3 or more failed attempts to perform a task) they do not understand, or inquiring as to their task.

H. Subjects will be informed of the purpose of the visualization that they are looking at, and then given control over the assembly to further explore the purpose of the design.

I. All of the subject's interactions with the program will be in a limited, ¼ screen size area, which will respond only to commands from the arrow keys and the mouse.

J. Participants will be thanked for their involvement at the end of their time.

K. The key topics of the conversation will be written down in order to collect feedback data (see Section C)

L. The feedback and data collected over the user study will be summarized and shared with the advisors of this MQP, Matthew Ward and Emmanuel Agu, as well as EMC personnel.

Appendix C - Sample User Study Questionnaire & Feedback

What did you like about the Chargeback visualization?

- Is the layout easy to understand?

  A little confused/ it doesn't seem connected.
- When selecting another section, do you appreciate the way the cylinder moves to correspond?

  No dependence from orange to purple.
- Do you like the LUN representation? Are the relationship lines helpful?

  Does the color have meanings? There is layout limitation. Try to avoid line crossing.
- Is it easy to navigate? Can you navigate back? Do you feel restrained? How so?

What did you dislike about the visualization?

Didn't think of scalability.

What would you suggest for improvements?

There might be different use cases. Need legend for color. Add fourth color-bar to group the application. The current design will disconnect with the terminology when gets 100-200 devices. How easy for the user to understand each attribute via the supply chain? Take orange and purple as blocks and make a triangle loop to represent the relationship between primary, local, remote.

What did you like about the Array visualization?

- Is the layout pleasing?
- When you highlight an array, does it give you the information you need? What about the information located on the left?
- Is it helpful to see the snapshot of the available arrays? Would you rather see more? Less?
- Do you enjoy the distortion effect? The mouse-over effect?
- Are the relationship lines easy to use and understand?

What did you dislike about the visualization?

What would you suggest for improvements?

What did you like about the Usage Over Time visualization?

- Is it easy to read? Can you tell what each graph represents?

What is the advantage of the circle? Concentric ring didn't add much to the visualization, and it was rather unusual.

- When you mouse-over the circular implementation, do you readily understand what it is showing you? When you mouse-over the trend graph, do you understand it?

- When you mouse-over the graphs, is the information visible on the left-side easy to read and understand? Is it helpful?

What did you dislike about the Usage Over Time visualization?

It's not useful. Looks like you are getting more capacity but didn't add values. There are no labels for times. Line charts seem to be easier.

What would you suggest for improvements?

Have two similar charts over, separate the axes. Just show percentage with trend line, and connect the two charts.

What did you like about the Usage by Value visualization?

- Press play on the slider. Watch it move through the functions. Then, select an individual time. Is this view helpful to your understanding of the information?
  No, it's confused.
- Mouse-over the visualization and look at the information. Does it give you all the details you were interested in?
  It's very helpful.
- Mouse-over the visualization and look on the left. Is the information represented helpful? What would you like?
  Yes.

What did you dislike about the Usage by Value visualization?

During play, all the changes are dynamic; it's hard to keep in mind for the user.

What would you suggest for improvements?

Drill down function can be added. Color the groups differently. Show the relative relationship between trend over time for different location.

What did you like about the User Interface?

- Are the legends helpful?
  Yes.

- Is the left-aligned design intuitive and easy to use?

  Very much.

- Are the visualizations implemented in a proper and easy-to-use manner?

  Yes.

Do you feel that the program gave adequate coverage of the information that it represented? If not, why?

- Seeking a way to show the relevant data without going too many steps.

Were any of the features not intuitive? Did you have difficulty with any of the visualizations? If so, please explain.