FUNDMATCH Project Literature Review

Timothy Gwynn
University of Cape Town
Rondebosch, 7701, South Africa
gwytim001@myuct.ac.za

ABSTRACT

The optimization of fund matching within large holding companies benefits both the holding company and its subsidiaries. In order to compare the results of different optimization algorithms it is necessary to create a method for visualizing the data produced by these algorithms. This review covers visualization techniques for resource allocation data as well as methods for the user-centered design(UCD) and evaluation of visualization models. Despite the lack of literature closely related to the subject topic it was found that for data of the type generated by fund match optimization a node-link graph was the most likely to be suitable. Additionally end-user expert evaluation was designated as the most practical form of evaluation for a project of this scale.

CCS Concepts

•**Human-centered computing** → **Visualization;** *HCI design and evaluation methods; Visual analytics; Visualization design and evaluation methods;* •**Information systems** → Enterprise resource planning;

Keywords

Visualization, User-centered design; Resource allocation; Finance; Holding Company

1. INTRODUCTION

Fund matching is a method large holding companies use to spread their financial resources between their subsidiary bodies. It primarily involves matching fund sources, profitable subsidiaries with excess finances to fund requirements, subsidiaries in need of external funding. These, intra-company loans are more cost effective than borrowing from external sources such as banks. It is important to optimize fund matching to minimize external funding needs across the holding company. However, many equivalent optimizations may be calculated, requiring expert knowledge to choose between them. For this reason the data needs to be visualized in order for financial experts to effectively and efficiently analyse it.

The data for this visualization consists of three types of node elements and their relations to one another. Each node has 3-4 dimensions of data directly associated with it. Fund requirements and sources both have a start date, tenor, amount and tax class. Balance pools, which are a special type of fund source, have an internal balance, tenor and allocation limit associated with each pool/node. These nodes will be connected to each other in such a way that the maximum possible number of requirements are fulfilled and sources and balance pools are used as efficiently as possible. This suggests a node-graph type visualization as an initial starting point. It will be necessary to display the multivariate data that forms each node as well as allow the user to traverse a node-graph like structure. Since much of the data is time orientated with sources and requirements appearing and disappearing over time it will also be necessary to show the relative time frame of the data being displayed to the user.

According to Keim et al. the majority of financial visualizations are two dimensional X-Y graphs that only display two dimensions of data. [9] If one considers the typical increasing line over time graph that is associated with the finance industry in media this is very apparent. While this is suitable to display the change in value of a single entity, assuming the value can be represented by a single variable, it is not useful for describing more complex systems. In cases where the data is inherently multi-dimensional it is necessary to use more advanced visualization techniques. This allows the user to gain a holistic view of the system while still being able to focus on specific components. This is according to the visualization mantra of Schneiderman et al. ”Overview first, zoom and filter, then details-on-demand”.[16]

The majority of financial graphics are generated using Microsoft Excel. This allows users to create a number of simple graphics such as line graphs, bar charts and pie charts. It is, however, unsuitable for creating multivariate visualizations.[13] It is possible for expert users to create somewhat advanced visualizations with the use of macros but this is beyond the capabilities of the majority of users and can often be done more easily using other tools.

Current financial visualizations are not suited to describe complex systems. They are too simplistic and generally lack interactive elements. Like many other graph-structured systems based on real world data there are a multitude of variables that must be displayed in the fund match visualization.[7] Using the current standard tools such as Microsoft excel it is not practical to create visualizations of such complex systems. It is essential to incorporate a variety of visual elements as well as interactivity in the visualization and break away from the low dimensional norm of financial visualizations.

It is therefore important to create an effective visualization that is capable of displaying multivariate data in a concise and efficient manner. To do this there are a number of guidelines and heuristics that should be taken into account when designing any effective visualization. These are summarized below and will form the basis of this project.

Since the data is multi-dimensional in nature it is essential to solve the multivariate visualization challenges outlined in Rheingans et al.[14] These are as follows: The spatial relationships between variables should be shown, different variable representations should not interfere with one another, joint distributions should be understandable and as many variables as can be effectively displayed should be shown. Doing this requires the use of multiple visual channels and variable differences should be represented by differing on one or more channels [22]. The visualization mantra of Schneiderman et al. should also be followed to facilitate this [16]

By displaying data visually expected patterns can be more easily observed and unexpected behaviours are more obvious.[10] Visualizations also allow for the comparison of data sets through a set of visual queries. By designing a visualization to support specific visual queries it is possible to have a large amount of relevant data on a single image. Adding interactivity to the visualization further enhances the number of visual queries that can be answered while avoiding creating a very busy graphic. An interactive visualization should allow for details-on-demand to be easily accessed such as by bringing up a detail window on a subsidiary node on the fund match graph when the node is moused over [5]

The aesthetic element of visualizations is not as essential as the other elements but should be considered nevertheless. An aesthetically pleasing graphic is often easier to understand and more pleasant to work with. There are a number of elements to consider when attempting to create an aesthetically pleasing graphic according to Tufte et al. [18]

The aim of visualization is to aid in the comprehension of large amounts of data by the user. It is therefore essential that when designing an effective visualization the user is integrated into the process as far as is possible. User-centered design is a philosophy that supports this integration of users in the design process[12] and there are a number of studies on various methods to effectively incorporate users in software design. Incorporating user feedback should be central to the iterative visualization design cycle.

Keeping in mind the aims stated above the body of this paper will explore modern visualization techniques more specifically related to our problem area. First we discuss methods for designing visualizations with a focus on UCD. We then analyse existing visualization conventions in the financial field. Next we focus our research on visualizations dealing with resource allocation especially where data changes over time and when the data is represented as a node-graph. This will be followed by a discussion of the strengths to adopt of each approach reviewed as well as weaknesses to avoid.

2. METHODS FOR DESIGNING VISUALIZATIONS

The design of an effective visualization is a non-trivial task. In part this is because it is hard to predict the most effective way in which the data to be visualized should be presented. As is common in many software projects the end user is often unsure as to what they precisely require or desire. Visualizations should therefore be created iteratively with feedback each iteration before the final product is decided on. [8] This feedback ensures the visualization can effectively fulfil visual queries while retaining aesthetic appeal to enhance usability.

2.1 User-centered design

User-centered design (UCD) is essential to create any sort of system that users can effectively and intuitively use. This is because a creator's intuition is often very different from the user's. It is hard to accurately predict how users will intuitively try to use the product. By focusing on UCD we put the user needs first and then design the product to fit these.[12] This requires more active involvement from users in the design but can lead to a much higher quality product in the end and avoids the creation of a unwanted product.

2.2 Design Principles

Norman et al. suggest a set of principles that are essential for effective design. One of these is to simplify the structure of tasks[12] which is the primary aim of visualization techniques. Norman et al. stresses that the user's memory should not be overloaded by trying to store large amounts of information at a time. Another principle that relates directly to visualizations is to get mappings correct, to ensure that the user understands the meaning of the graphics displayed and intuitively knows the gist of the information being displayed. These principles cannot be followed without UCD, one cannot predict the correct mappings without asking the end user what their mappings would be and even then the end user may be unable to adequately explain.

2.2.1 Users in the design stage

It seems that identifying the users of a system is a relatively straightforward task but it is not necessarily as simple as it first appears.[1] Besides the end users there are a number of stakeholders in any system that should be considered as users in the UCD approach. These include the managers of the end users who require a certain level of productivity/ efficiency from their teams as well as those who are affected by the use of the system i.e. third parties. These other types of users must be taken into account during the design process together with the end users. Although they do not necessarily need to have a representative on the design team their needs should be taken into account when designing a system.

Once users have been identified the design process can begin to try to develop a number of possible solutions to their needs.[1] In the early phases of design it is necessary to maintain a high level of flexibility in design such that any changes identified as necessary can be identified and corrected quickly and cheaply as well as to allow for a number of alternative designs to be tested without large expenditures of resources. A popular method of doing this is to use paper prototypes in the initial phase. These allow for fast and cheap prototyping while providing users with a good initial idea of the look and feel of potential designs. This phase may also produce information on needs that users did not identify at first when describing their problem.

Abras et al. suggests that there are a number of UCD techniques that can be ordered by where in the design cycle they are useful.[1] Before the design cycle starts it is necessary to identify the needs and expectations of users in order to conceptualize the product. Early in the design cycle the focus is on getting data from the users on how the users wish the system to be used. Methods for doing this include work interviews and questionnaires, focus groups and on-site observations. While each of these techniques produce different information the context for all of them is to identify how the system will be used. After this data has been gathered designers will create a number of prototypes on which users will perform simulations in order to evaluate a variety of solutions. These can range from simple paper prototypes discussed above to fully functional systems which are close to being deployable. The final stage of the design cycle includes usability testing as well as a final round of interviews and questionnaires to establish whether users are satisfied with the final product.

2.3 Evaluation methods

2.3.1 Tiered User Testing

User testing is an essential component of UCD. In their paper on UCD and evaluation Hix et al. break user testing into distinct categories; Heuristic evaluation, Formative evaluation and Summative evaluation.[8] These categories are listed in order of increasing cost and all serve distinct purposes at different points in the evaluation process. Hix et al. carried out a number of rounds of evaluation of a real-time battlefield visualisation using these methods the results of which are explained in detail below.

Heuristic evaluation is carried out by a number of user interaction design experts. In Hix et al.'s experiment these experts would work independently or together in order to evaluate the interactive elements of the battlefield visualization. After a number of cycles of heuristic evaluations Hix et al. were able to identify and address a number of gross issues with the initial system being tested.[8] Since the user interaction design experts were not necessarily knowledgeable in the problem area it is unlikely that they would pick up any subtle, contextual issues but they could pick up the most obvious flaws in the system at a fraction of the cost of full scale user testing since only a small team of experts is needed.

Formative evaluation is carried out by a group of novice users who are introduced to the system and then given a series of tasks to perform. In their experiment Hix et al. were able to collect both quantitative data, such as the time taken to complete certain tasks as well as qualitative data in the form of user comments.[8] These formative evaluations also allowed critical incidents in the user experience, often related to user errors. According to Norman et al.'s principles of design it is important to highlight and inform the user when these incidents occur and to provide them with a path for correction.[12] While Formative evaluation requires a larger number of users to provide enough data points for valid statistical analyses of the quantitative data it also can highlight design issues with greater precision than the evaluation of a small number of experts. It also more accurately models the actual use of the system and therefore may highlight issues missed by experts due to incorrect assumptions about the typical user's level of knowledge. Hix et al. found that the previous heuristic evaluation reduced the need for a large number of formative evaluation cycles reducing the resource cost of this evaluation stage. [8]

Summative comparative evaluation seeks to compare the new design to similar existing systems using empirical assessment.[6] It is normally performed using close to final versions of the new design and focuses on quantitative data. It requires there to be some definition of performance of the system such that when the new system is compared to the old system one can be defined as better than the other. Due to the size of such a study and the need for a fully functional version of the system this type of evaluation has a high resource cost. Hix et al. found that their previous evaluations ensured that the systems being compared were similar enough to ensure the new system was comparable to the old in terms of both function and quality and that a summative evaluation was therefore justifiable.

In their expirement Hix et al. showed that using an iterative process with a number of evaluations of different types was useful to save both time and money and was more effective than using purely summative testing at the end of the design cycle.[8] This is a valuable result as one of the biggest drawbacks of UCD is the large cost of involving users in the design process. By revealing usability problems as early on and as quickly as is possible it is possible to minimize this cost while retaining the benefits of UCD.

2.3.2 Case Studies

Case studies involve reporting on the use of an artififact in the typical work conditions and environment that an end user would use the artififact in.[19] Valiati et al. carried out a number of case studies to evaluate visualization tools. These case studies involved a small number of users who were motivated to use the tools for their own purposes. Users participated in a number of sessions during which they were aided by experts in using the visualization tool. Feedback was gathered from letting the user think out loud during these sessions and having an observer record these thoughts along with other details of interest. Valiati et al. found that these case studies resulted in a high quality understanding of the quality of the visualizations being tested. They also emphasised that having experts in the tool work alongside the novice users performing the evaluation was essential for effective evaluations.

2.3.3 Expert Reviews

In their 2005 paper on evaluating visualizations Tory et al. explore whether expert evaluations of visualizations are effective.[17] They used small numbers of HCI experts to evaluate visualizations. These experts were asked to carry out heuristic evaluations in order to identify problems that might arise using a interactive visualization. They then carried out a similar experiment using end-user experts as well as HCI and graphic experts to evaluate a different visualization. Their conclusion was that expert evaluation was effective for detecting a number of issues with the visualization but should not be the only type of evaluation as HCI experts may not predict all possible end-user actions. They also found that by including experts in the end-user field they received a higher quality of feedback than when using purely HCI experts. Even though expert reviews do not pick up on all issues they are valuable since, compared to large user studies, they are very cost effective. This means that smaller projects that are unable to afford the resources necessary to conduct full scale user testing can choose to use expert reviews as a cheaper alternative.

3. VISUALIZING RESOURCE ALLOCATION

Fund matching involves linking fund sources or balance pools, a special type of source, to fund requirements. The goal is to optimise this according to a set of goals. Briefly these goals include; matching sources and requirements of the same/similar tenor, the lifetime of a source/requirement, to each other; to minimize wastage and minimizing the amount of external funding necessary to satisfy all requirements. This means that fund matching is a type of resource allocation where sources and balance pools are resources and requirements are the entities to which these resources must be allocated.

Since the resource allocation algorithm may produce a number of closely optimal solutions it is essential that a human analysis is possible on the possible solutions produced. This allows for experts in the field to make contextual judgements that the optimization algorithm cannot due to the simplification of the data. Since human interaction is an essential component in visual analytics it means that the visual interface with the data must be both intuitive and efficient.[10]

In this section we will analyse methods and techniques that have been used to visualize the data produced by resource allocation processes. We start with a review of a system used to visualise a fund manager which incorporates data very similar to that which the fund match visualization will display. After this we analyse a number of ways in which dynamic graphs can be visualised along with the context in which each of these methods are suitable. Following on we discuss two case studies on the subject of the design of node-graph visualizations, these further explore some of the techniques discussed in the *dynamic graphs* section. We also explore the importance of representing time in visualizations which is an essential component of dynamic graphs. Finally a number of techniques for dealing with high-dimensional data in visualizations is discussed.

3.1 3D Fund Manager Visualization

While Fund Managers are not exactly the same as fund matching they exhibit many of the same attributes. Namely they are made up of a time dependent network of objects that can be represented as a node-graph.[4] Dwyer et al. created two variations of 3D graphs with the aim of visualizing the behaviours of Fund managers. A note to make is that unlike many other stock market visualizations Dwyer et al. knew the details of which bodies where involved in each transaction, this is conveniently similar to our case where all subsidiary details are known.

In their visualization Dwyer et al. use the plane dimension to place nodes relative to each other by group nodes of high connectivity to each other and minimizing crossing edges.[4] The vertical dimension was used to represent time. By using this arrangement with both perspective and parallel views it is possible to see the distinct edges between nodes even when there are multiple edges between the same pair of nodes. The relative time at which the edges where created is also easily identifiable.[4] Since the time periods were discrete this allowed for the vertical dimension (and therefore the node heights) to be broken down into sections. Dwyer et al. used this to create tapering sections in their node columns where each section was a time period and the width at the top and bottom of the section was based on the share price of the fund manager represented by the node at those times. This leads to ease of identifying which edges are made in which time period as well as adding an extra dimension of information. See figure 1 for an example of this visualization.

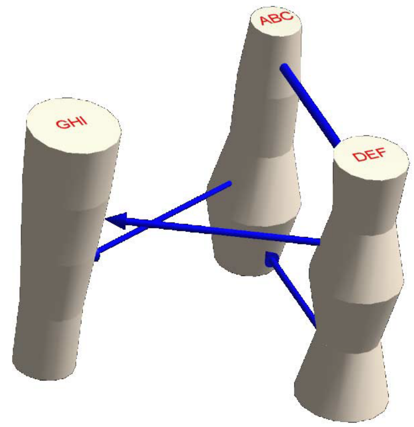

Figure 1: 3D Column Fund Manager Visualization[23]

Dwyer et al.'s second variation involved adding a concept of *closeness* to their data and replacing the vertical columns with worms that had an added degree of freedom such that they were able to bend away from the base of the node they represented.[4] This *closeness* allowed Dwyer et al. to show how involved nodes were with one another depending on the edges between them at a point in time. An example of this version of the visualization can be seen in figure 2.

Dwyer et al. explored a number of interesting ideas in their paper and their visualization techniques had both strong and weak points.[4] At first glance it seems that the entire plane dimension is wasted as it does not display any explicit dimensions of data. However, by displaying clustering effectively as well as preventing clutter and crossed edges important data is made visible by this decision. It is important to learn from this that one should not neglect visual queries that are not linked to specific data dimensions and that showing the relationships between entities is equally as important as the properties of the entities. However, there are a number of issues with the visualizations produced by Dwyer et al. The biggest of these being the neglect of a variety of visual channels. Color and texture is not used at all,

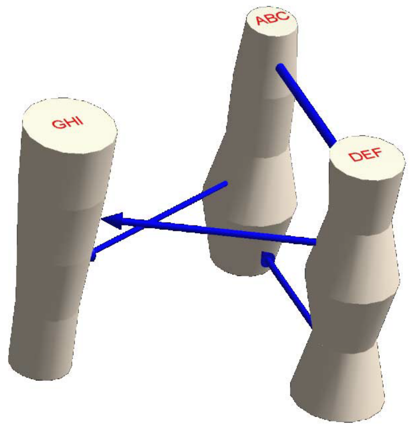

Figure 2: 3D Worm Fund Manager Visualization[23]

there is little differentiation in shape and no symbols are in use either. Using a two colour pattern to stripe time steps would have made it much more obvious than it currently is even with the tapering sections. Fund managers are labelled with 3 letter acronyms at the top of each node which may require a large amount of knowledge recall, it might have been better to replace these with the relevant fund manager logos if these are well known to users of this data.

The most unique aspect of Dwyer et al.'s visualization was the use of 3D. While it can be effective in certain circumstances 3D often obscures more than it reveals.[4] However, studies on using 3D in node-link graphs suggest a significant improvement in the complexity of the graph that can be "read" by a user.[23] In the case of this visualization the 3rd dimension is effectively used to display both the variable of time but also to show changes in node qualities over time. This is a considerable gain and allows much additional information to be shown in a single graphic. The largest issue with the use of 3D in this visualization is the need to switch between parallel and perspective views which may hinder visual queries that depend on information displayed in both views separately.

3.2 Dynamic Graphs

Dynamic graphs are used to represent the relationships between a number of entities and how these relationships change over time. The nature of the data in the fund match visualization is of entities being connected to one another for periods of time before disconnecting. Therefore this section reviews a number of different methods for visualizing dynamic graphs in order to discover what would be most effective for the fund match visualization.

In their paper *The State of the Art in Visualizing Dynamic Graphs* Beck et al. explore how different variations of dynamic graph visualizations can be broken down and how the separate approaches differ from one another.[3] According to Beck et al. Dynamic graphs primarily differ in their method of displaying changes over time, either by animation or using some form of time line.

Animation techniques for dynamic graphs can be subdivided into general and special purpose layouts. [3] General purpose layout graphs allow for the display of any type of graph. The shape of these graphs can either be fully calculated prior to the animation (off-line animation) or can be adjusted as the animation is played (online animation). Online animation is suited to data sets where future graph states are not necessarily known. Yee et al. created an online animation graph using a radial layout.[25] This allowed them to pick a central node for each new graph state around which the graph could be arranged. The graphs were then animated from one state to the next to represent the passage of time. Special purpose layouts are used in the case where specific characteristics of the graph mean that specialized layout approaches are either required or greatly beneficial when displaying the graph.

Time-line techniques, which use a time-to-space mapping, can be broken down into node-link based representations or matrix representations.[3] Node-link based representations can be formed by a series of juxtaposed node-link graphs laid out along a linear time-line or by super-imposing the nodes on top of each other. The super-imposition technique lends itself towards 3D node-link visualizations where the z-dimension represents time such as the fund manager visualization discussed above.[4] An alternative visualization technique is an integrated approach where the time-line is intrinsically linked into the visualization such that each edge represents the flow of time. These approaches to node-link based representations of dynamic graphs are not exclusive and may be combined in some manner to fulfil certain requirements of the graph layout. Matrix based representations fall into two categories; either those using intra-cell time-lines where each cell in the matrix contains an individual time-line or those made up of a number of juxtaposed matrices each which corresponds to a time step.

3.2.1 Node-graph Visualization techniques

Two essential components of all node-graph visualizations are links between nodes and the arrangement of nodes relative to each other. While these components are essential there are a number of ways in which to visualize them, some of which are discussed below.

Xu et al. performed a user study on the shape of edges in a node-graph visualization to determine what type of edge, straight or curved, was most effective both from an reading accuracy and aesthetic perspective.[24] They found that users performed significantly better at reading graphs with straight edges over graphs with uniformly curved edges and that users found the straight edges more aesthetically pleasing prior to performing the performance tests in their experiment. Examples of both types of graph can be founf in figures 3 and 4. They did note that a combination of straight and curved edges where curved edges were used only where necessary did not have a significant effect on user performance when compared to graphs with only straight edges. It is likely that users performed worse on the graphs with curved edges as they included a much higher number of intersecting edges which makes it significantly harder to follow paths over the graph. This is reinforced by the fact that the larger the curves, and therefore the higher the number of intersections the worse users performed.

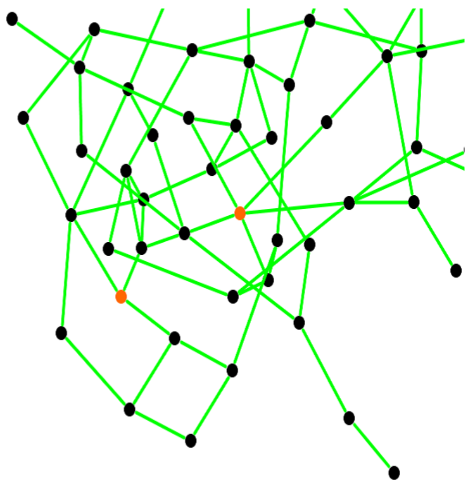

Figure 3: Straight-edged node graph Visualization[24]

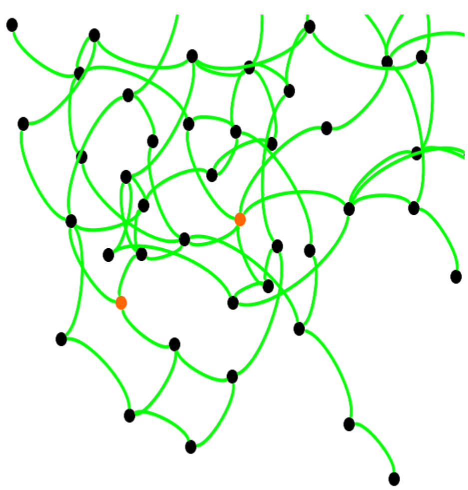

Figure 4: Curved-edged node graph Visualization[24]

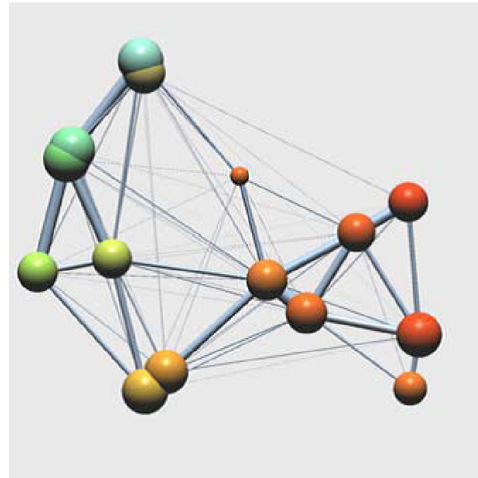

Figure 5: Node Graph with collapsed nodes[20]

In their paper *Interactive Visulaization of Small World Graphs* Van Ham et al. discuss how to effectively facilitate clustering in interactive visualizations of node-graphs.[20] Clustering allows the user of a visualization to identify which sections of data are both closely and loosely linked to one another, it also reduces the complexity of the visualization and the number of intersecting edges. By using interactive visualizations it is possible to go further and at certain levels of detail condense a group of tightly clustered nodes into a single entity further simplifying the graph and reducing the total number of edges for the user to follow. An example of this is shown in figure 5. In their paper Van Ham et al. suggest a number of methods for clustering and scaling data in such a way that an useful level of detail is always visible while preventing unnecessary clutter in the graphic.

3.2.2 Visualizations over time

According to Aigner et al. time should have a special case of consideration in visual analytics techniques, primarily because of the high degree of interactivity users often wish to have with the time dimension.[2] In this section we explore some of the factors to account for when visualizing time.

Data is often represented at different levels of temporal granularity with visualizations needing to display both long and short term events. This lends itself to incorporating multiple views into the visualization with each view corresponding to a different level of temporal granularity.[2] This can also be linked to scaling the visualization with respect to time and level of detail. Allowing the user to zoom in on both an area on the graph as well as a specific window in time means that fine detail between closely associated elements in a short time window can easily be examined.

Another important aspect of time-orientated visualizations to consider is whether the final data points are known. In the case where this is not true it can become significantly more complex to determine the layout of the graph in order to allow for clustering and minimize intersecting edges.[3]

3.3 High Dimensional Data Visualizations

In the paper *A Survey of Graph-Based Representations* *and Techniques for Scientific Visualization* Wang et al identifies one of the remaining challenges in graph visualization as graph simplification.[21] Dynamic graphs already do this to some degree as they seek to seperate time out into its own dimension thus preventing data from different points in time obscuring other data. However, even within these time steps it is essential to simplify the graph as far as possible while retaining all the information that is needed, both to prevent the user being overwhelmed by a single graphic and to facilitate comparisons between different time steps.

Dimension reduction is a method to reduce the number of dimensions used to represent the data.[11] There are a number of different approaches to do this but the principle of collapsing data dimensions into each other remains the same. Linear Projection involves projecting high dimensional data into a low dimensional space using a linear transformation. This can be combined with a number of different views in an interactive visualization to allow the user to access the full array of data dimensions while keeping individual graphics relatively simple. Non-linear dimension reduction can be used in either a metric or non-metric setting. The metric setting is best handled by graph-based techniques where clustering algorithms can be used to reduce the data dimensionality. When Data sets are extremely large and complex linear and non-linear projections may not be sufficient an a technique know as control points based projection can be used where the user can select specific control or anchor points before performing the dimensionality reduction. This allows for greater user control of the dimensional reduction as well as scaling better with big data.

Subspace Clustering is a method that combines the concepts of traditional clustering techniques, dimension reduction and interactivity.[11] It allows the user to explore clusters of either dimensions or data points in order to identify certain patterns in the data. Dimension space exploration, a subspace clustering technique, allows the user to interactively group dimensions into subsets to explore the relationships and shared patterns between them. This can be combined with data point clustering to give the user further flexibility to explore dimension and point embedding simultaneously. Subspace clustering can also involve automatically creating dimension subsets rather than requiring the user to manually create these. This is mainly effective when dimensions are not tightly coupled to one another. There are a number of different algorithms that provide subspace clustering services. In their 2010 paper Sembiring et al. present an algorithm known as Proculus which allows the user to cluster entities according to a specified variable.[15] Figure 6 shows an example of this algorithm in use where points of the same colour would be clustered.

Once the data has been simplified by the methods discussed above among others it is still necessary to map the data sets to a visual.[11] There are many approaches to this and the best approach or combination of approaches to use is determined by the data that needs to be visualised. However, some visual mapping techniques can be used in almost all visualizations of high dimensional data. One of these is a hierarchy-based approach. The exact nature of the hierarchy is data dependant as it is possible to have either a dimensional hierarchy or topological hierarchy as well as

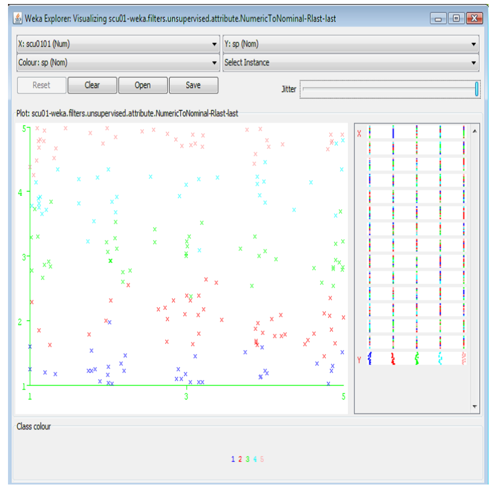

Figure 6: Example of a PROCULUS in use, nodes of the same colour form a single colour[15]

other variations in specific circumstances. This is particularly important in interactive visualizations where the user can zoom in on data in order to view details, whether dimensional or topological, that are further down the hierarchy than is shown from a top level overview.

4. DISCUSSION

In the section on visualizing resource allocation we examined visualizations with similar data properties to the fund match data as well as cutting edge techniques to visualize data of this format. While there where some useful techniques included in the fund manager visualization it also had a number of weaknesses. Additionally the use of 3D techniques is less appealing in the fund match visualization as there should never be more than one link between the same two nodes. In the dynamic graphs section node-link representations appeared to be the most suitable for the fund match data. Animation techniques are promising but may hinder comparison between different complete solutions to the optimization problem as they do not support a global overview. Matrix based representations seem better suited to smaller sets of data as well as data with a static number of nodes which is not the case in the fund match project. Based on Xu et al.'s research it is also obvious that nodes should be linked using straight lines. Additionally the raph should be arranged in such a way that intersecting edges should be minimized. With regard to the clustering techniques it is apparent that this area will require some work, however the complexity of this problem will depend on how large the actual data is. This also applies to the issues reviewed in the high-dimensional data section as the actual complexity of the data as well as the exact nature of visual queries will influence whether it should or an be simplified.

In the section on methods for designing visualizations various evaluation methods where discussed and UCD was explored in depth. The evaluation methods reviewed in this section had a number of qualities that made them distinct from each other. However, the time and resource constraints of this project must be taken into account. Because of these it is not feasible to carry out a full set of tiered user evaluations due to lack of both resources and time. Case studies are a promising option as they suit the context in which the fund match visualization is being created. Unfortunately due to the intensive nature of these studies it seems unlikely that time constraints will allow for these to be effectively carried out. Although they do not provide evaluations of the same completeness of the previously mentioned techniques it seems that end-user expert evaluations are the most practical option for this project.

5. CONCLUSIONS

There is unfortunately very little academic literature relating directly to fund matching. This is also true for visualization techniques of financial data in general. Because of this we were forced to focus this review on visualization models that matched the data format as well as any that had similar context.

From this review it seems that a node-link graph will be suitable to visualise the fund match project data. However, the exact nature of this visualization will depend on getting a list of visual queries as well as an actual data sample. Additionally, it is apparent that the most practical form of evaluation is end-user expert evaluation. This too is subject to review as it is still unknown what human resources will be available for testing.

6. REFERENCES

[1] C. Abras, D. Maloney-Krichmar, and J. Preece. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4):445–456, 2004.

[2] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented dataâĂŤa systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

[3] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. *EuroVis STAR*, 2014.

[4] T. Dwyer and P. Eades. Visualising a fund manager flow graph with columns and worms. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 147–152. IEEE, 2002.

[5] S. G. Eick. Engineering perceptually effective visualizations for abstract data. In *In Scientific Visualization Overviews, Methodologies and Techniques, IEEE Computer Science*. Citeseer, 1995.

[6] J. L. Gabbard, D. Hix, J. E. Swan, et al. User-centered design and evaluation of virtual environments. *Computer Graphics and Applications, IEEE*, 19(6):51–59, 1999.

[7] S. Hadlak, H. Schumann, and H.-J. Schulz. A survey of multi-faceted graph visualization. In *Eurographics Conference on Visualization (EuroVis)*, pages 1–20, 2015.

[8] D. Hix, J. E. Swan, J. L. Gabbard, M. McGee, J. Durbin, T. King, et al. User-centered design and evaluation of a real-time battlefield visualization virtual environment. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 96–103. IEEE, 1999.

[9] D. A. Keim, M. Ankerst, and H.-P. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of the 6th conference on Visualization'95*, page 279. IEEE Computer Society, 1995.

[10] A. Kerren and F. Schreiber. Toward the role of interaction in visual analytics. In *Proceedings of the Winter Simulation Conference*, page 420. Winter Simulation Conference, 2012.

[11] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. In *Proc. Eurographics Conf. Visualization*, pages 20151115–127, 2015.

[12] D. A. Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[13] K. Pantazos and S. Lauesen. Constructing visualizations with infovis tools-an evaluation from a user perspective. *GRAPP/IVAPP*, 2012:731–736, 2012.

[14] P. Rheingans and C. Landreth. Perceptual principles for effective visualizations. In *Perceptual Issues in Visualization*, pages 59–73. Springer, 1995.

[15] R. W. Sembiring, J. M. Zain, and A. Embong. Clustering high dimensional data using subspace and projected clustering algorithms. *arXiv preprint arXiv:1009.0384*, 2010.

[16] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[17] M. Tory and T. Möller. Evaluating visualizations: do expert reviews work? *Computer Graphics and Applications, IEEE*, 25(5):8–11, 2005.

[18] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[19] E. R. Valiati, C. M. Freitas, and M. S. Pimenta. Using multi-dimensional in-depth long-term case studies for information visualization evaluation. In *Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, page 9. ACM, 2008.

[20] F. Van Ham and J. J. Van Wijk. Interactive visualization of small world graphs. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 199–206. IEEE, 2004.

[21] C. Wang. A survey of graph-based representations and techniques for scientific visualization. 2015.

[22] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.

[23] C. Ware and P. Mitchell. Visualizing graphs in three dimensions. *ACM Transactions on Applied Perception (TAP)*, 5(1):2, 2008.

[24] K. Xu, C. Rooney, P. Passmore, D.-H. Ham, and P. H. Nguyen. A user study on curved edges in graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2449–2456, 2012.

[25] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *InfoVis*, page 43. IEEE, 2001.