Space-filling Techniques in Visualizing Output from Computer Based Economic Models

Richard Webber[a], Ric D. Herbert[b] and Wei Jiang[bc]

[a] National ICT Australia Limited, Locked Bag 9013, Alexandria, NSW 1435, Australia
Email: Richard.Webber@nicta.com.au
[b] School of Design, Communication and Information Technology, The University of Newcastle
University Drive, Callaghan, NSW 2308, Australia.
Email: Ric.Herbert@newcastle.edu.au; and Wei Jiang: jackie4my@yahoo.com.cn
[c] Shanghai E&P International, Inc.
12 & 37F, Shanghai Information Tower,
211 Century Ave., Pu Dong New District, Shanghai 200120, China

Abstract: One important factor concerning economic models is that frequently large amounts of data are produced. There is the research issue of how end-users (who may not be researchers or model developers) can be presented with this data so that maximum benefits can be attained from the data production. The usual approach with economic models is a series of tables or data series plots. In this paper we use space-filling information visualization techniques as an aid to user's understanding of data from an economic model.

Based upon evaluation of the effectiveness of existing treemap and sunburst techniques through user experimentation, we introduce two new space-filling visualization techniques. We also describe fisheye-lens techniques applicable to these new visualizations.

Keywords: Information visualization; Agent model; Minority Game; Treemap; Sunburst.

1. INTRODUCTION

One important factor concerning economic models is that frequently large amounts of data are produced. There is the research issue of how end-users (who may not be researchers or model developers) can be presented with this data so that maximum benefits can be attained from the data production. The usual approach with economic models is a series of tables or data series plots.

A limitation of tables and plots is that they are intended for displaying the values of a single attribute – that is, they are one-dimensional. Multi-dimensional plots (showing many attributes at once) are possible, but only by placing several one-dimensional plots in parallel (see Figure 1 [8] for an example).

More advanced information visualization techniques [8] are needed to show more complex relationships between multiple data attributes. This is not to say that advanced information visualization techniques can replace traditional presentation of data with tables and plots, but rather that they can compliment these techniques and aid the end-users understanding of the data.

Interactive computer displays that allow the end-user to manipulate data can aid the end-user in understanding the data. Whereas tables and plots were originally designed for hard-copy output, information visualization techniques are often designed to incorporate the interactive manipulation of data. Such techniques as zooming in and out, providing additional data with mouse-over and selection of a subset of the visualization components are frequently incorporated into the information visualization software. These techniques are included with the aim of aiding understanding of the data.

Many economic models produce massive quantities of data, and trying to understand the interactions between the data and 'what it all means' is not a simple task. Often the end-users of the data are not the model developers – they may be policy-makers who are trying to understand what this data means. We ask the question of whether information visualization can aid in the understanding of data from economic

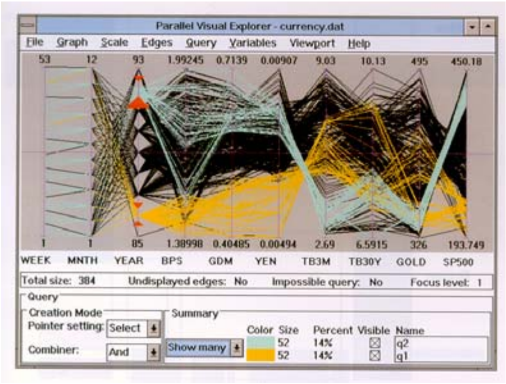


Figure 1: Parallel coordinates visualization [8]

models.

In this paper we consider a particular class of information visualization techniques: space-filling information visualizations; and we use data from a particular economic model: the Minority Game. In [5] we applied existing space-filling visualization techniques to the output of the Minority Game. In [4] we evaluated these techniques using experiments on end-users' understanding of financial markets modelled by the Minority Game. In this paper we present new space-filling techniques designed to overcome shortcomings of the existing techniques identifier by these experiments.

2. THE MODEL

The Minority Game [1][2] is a game consisting of several rounds where an arbitrary odd number of players choose one of two possible options each round. The two options could be zero or one, A or B, buy or sell, or any two things indicative of a competitive situation where there are two possible options the agents involved can choose from at any given point in time.

Each round of the game results in a majority of players choosing one of the options. The players who did not choose the majority option are in the minority and are the winners of that particular round – hence the name Minority Game (if all players chose the same option then there were no winners that round). Each player has a score, and their score increases when they win a round.

The number of winning players each round can range from 0 to $(n-1)/2$, for a game involving n players. Each of the players can remember the winning option (the option chosen by the minority of players) for the previous m number of rounds. This m represents the memory size of each player, and is fixed for all players.

Strategies are central to the behavioral dynamic exhibited by the Minority Game. In addition to each player being able to remember the previous m winning options, they also have a set of s strategies, each of which determines what option they should choose for the current round based on the history of previous winning options. For m rounds there are 2^m possible winning histories – thus each strategy has a size of 2^m (one bit representing the option to choose for each of the possible winning histories).

The actual winning history is known as the signal. The length of the signal is normally much greater than m. However, only the m previous results are considered by the players. Like players, strategies themselves also accumulate points. If a player's current strategy enabled them to win the current round its score increases. A strategy is deemed to be the current strategy of the player if it has the greatest score, with ties between best scores (and initially, when all strategies have no score) being broken arbitrarily. If a player possesses a strategy that is not the player's current strategy but would have won the current round had it been, its score also increases.

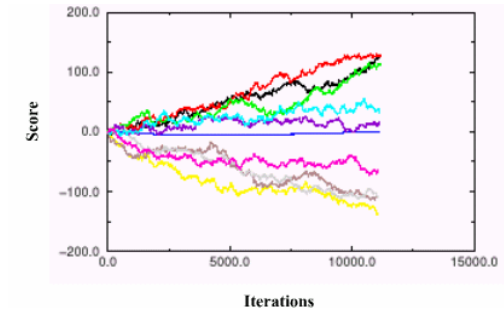


Figure 2: The scores of agents over time [2]

There are many variations on the Minority Game. We consider players to be agents (companies) in a financial market. Their score represents their capital stock. In our version of the game evolution occurs periodically with losers being replaced with clones of those agents with the largest capital (the winners).

From this model much data can be produced, especially when there are a large number of agents. There is the signal, each agent's score after each round, the 'virtual' score for successful strategies, and the evolutionary component of death and cloning of agents.

3. VISUALIZATION

The usual visualization of data from the Minority Game is in a time plot at each round, such as Figure 2 [2]. It may be of the signal, number of winners, average capital stock, etc. Yet often users are concerned with other questions. For example: do the winners follow similar strategies? What is the effect of evolution on the signal?

The data from the Minority Game can be stored or represented by a tree data structure. Nodes can store data such as capital and strategy, and the edges can represent the evolutionary history of the agents.

We use such a data structure to store data at each evolutionary step. At each step, the old agents become the ancestor agents of the new agents. For example, in Figure 3 there are 3 agents in time t1. When evolution occurs, say agent A is the agent with the highest score. It duplicates itself as A* in

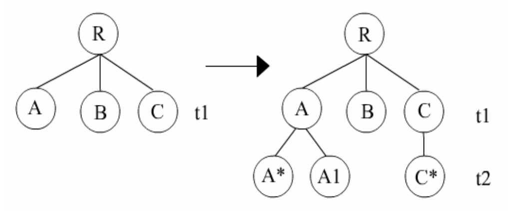
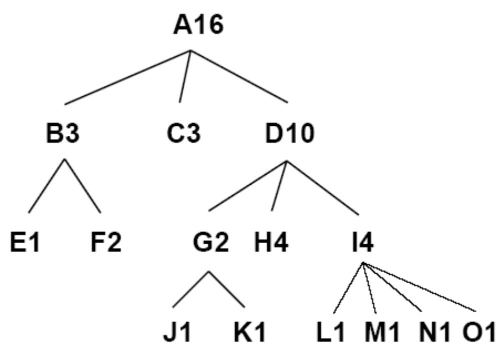
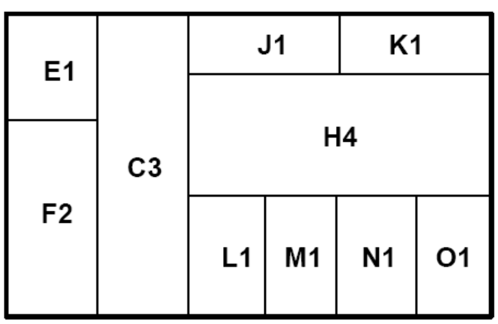


Figure 3: The evolutionary tree of a Minority Game



(a) Tree data structure



(b) Treemap visualization

Figure 4: A tree data structure and its treemap visualization [11]

the next time step t2. Agent B, who has the lowest score, is removed out of the game. To retain the same number of agents in the model, a new agent A1 is created as a child agent of the best agent A. Agent C is neither the best nor the worst one, so it only duplicates itself in time t2. The root node is the time before the game starts and the leaf nodes are the final evolutionary time-step.

Previously [4][5], we have used the treemap [7][12] and sunburst [9][10] space-filling techniques to visualise the game. A treemap uses a rectangular subdivision of space to represent a tree data structure. Figure 4 [11] gives an example of the conversion of a tree data structure into a treemap. More details of treemaps can also be found on the treemap Web site [12].

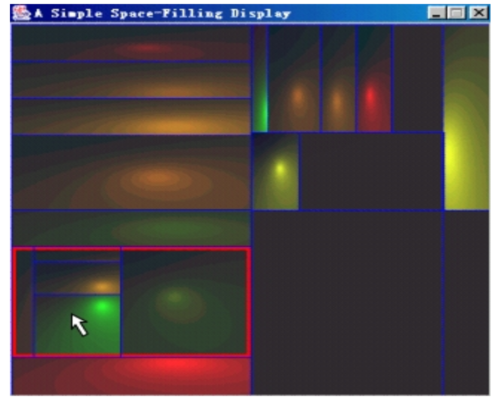


Figure 5: A treemap visualization of a Minority Game

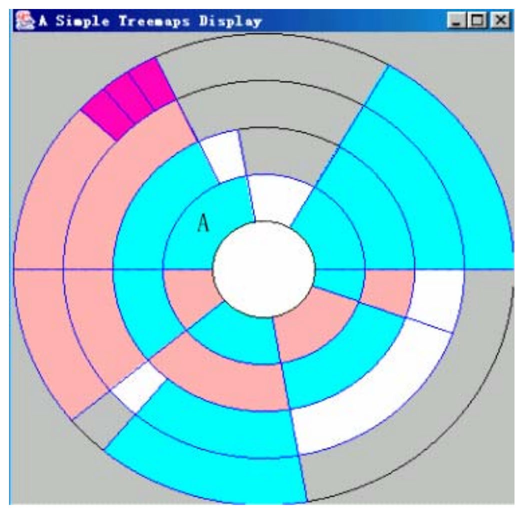


Figure 6: A sunburst visualization of a Minority Game

We used a squarified treemap in an attempt to overcome the issues of long thin rectangles appearing in the visualization. We also used a cushioning technique [11] to help the end-user to follow the evolutionary hierarchy. Finally, we used fisheye-lens techniques [3] so that the user could zoom into areas of the diagram.

For the treemap we used size of rectangle to indicate the agent's score (capital stock) and used color to indicate strategy. Similar colors indicate similar strategies. An example of a treemap used to visualize a Minority Game is given in Figure 5.

Sunburst [9][10] is a radial space-filling visualisation technique. It represents hierarchical data structures by separating space into a set of homocentric circles. The area between two adjacent circles represents one level in a tree structure. The circle in the centre represents the root node, with the hierarchy moving outward from the centre. Each circle can be separated into

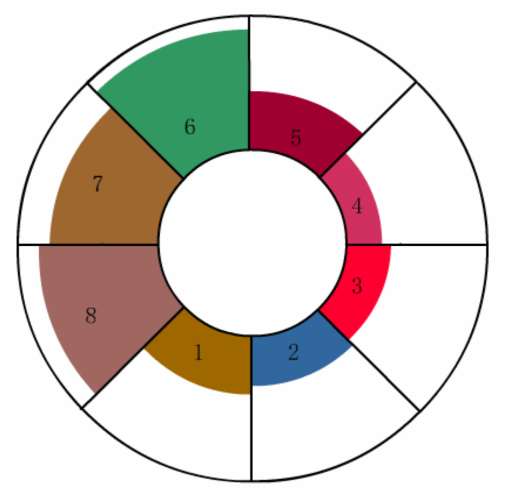


Figure 7: A filled sunburst

segments by radial lines. Each segment represents a node of the hierarchy. Child nodes are drawn within the angle occupied by their parent node. For more details see the sunburst Web site [9].

Similar to treemap, for the Minority Game visualization we used segment size to represent capital and color to represent strategy. An example of a sunburst used to visualize a Minority Game is given in Figure 6.

In [4] we conducted usability experiments on using treemap and sunburst to visualize data from Minority Game simulations. From these experiments we drew the following conclusions:

- treemap is more effective than sunburst for comparing the capital of companies (based on node size);
- treemap is more effective than sunburst for comparing the strategies of companies (based on node colour); and
- the effectiveness of cushioned-squarified treemap is similar to that of sunburst for understanding the evolutionary hierarchy of a financial.

Usability experiments on the fisheye-lens techniques found that these techniques aided users in understanding large data sets, but hindered users in understanding small data sets for both treemaps and sunbursts.

Interestingly, a subjective questionnaire revealed that users prefer sunburst over treemap.

4. NEW SPACE-FILLING TECHNIQUES

The main contribution of this paper is to present two new techniques that extend sunburst, exploiting end-users' subjective preference for this technique, while incorporating the objective benefits of treemap. In particular, these new techniques attempt to overcome sunburst's shortcomings when comparing the capital of companies based on node size, and for the visualisations to make use of the available (rectangular) screen-space.

The first of our new techniques is called a *filled sunburst*. A filled sunburst is a radial space-filling visualisation technique, much like the traditional sunburst. However, in a filled sunburst all companies at the same level in the evolutionary hierarchy have the same angular size. To represent a property of the companies (such as their capital), a proportion of each segment is filled with colour from the inner circle towards the outer, as shown in Figure 7 – the size of this proportion represents the value of the property. As with the traditional sunburst, the colour used to fill the segment can represent another property of the companies (such as their strategies).

This technique overcomes an issue of both sunburst and treemap in that the capital of companies can easily be compared across *all* companies, not just those companies which are children of the same parent in the evolutionary hierarchy. To further enhance this capability, users should be allowed to "mark" the visualisation with concentric circles that show the size of the filled region for one or more companies – all other companies can then be easily compared against these marks. This is illustrated in Figure 8.

One drawback of filled sunburst is that, if a company has a particularly low capital value, then its colour may not be clearly visible and its strategy cannot be easily determined. This problem can be partially overcome by applying

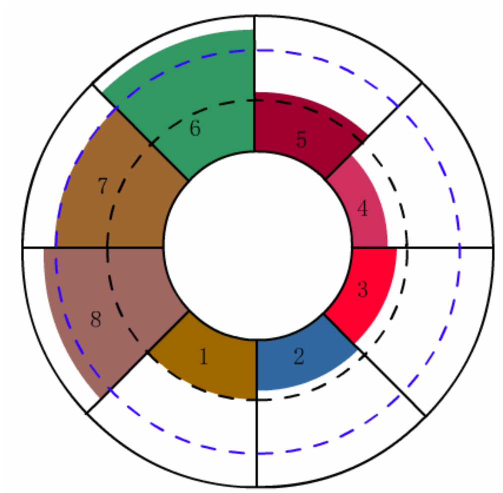


Figure 8: A filled sunburst with "marking"

fisheye-lens techniques to the filled sunburst visualisation. For example, users can scale the filled proportion of companies in a visualisation, allowing smaller companies to become visible. Such scaling can be applied to all companies at once or to only those companies in a single level of the evolutionary hierarchy. Scaling can be linear (preserves the capital ratios between companies), logarithmic (shows the most detail for small companies) or even histogram-based (categorizes companies based on their capital values). If the scaling causes the filled proportion to be greater than the segment, then a special color can be used to indicate that the company's capital is greater than the maximum representable value.

Alternatively, instead of increasing the proportions of the segments which are filled, users can chose to increase the size of the segments themselves by changing the radii of the concentric circles which make up the filled sunburst visualisation. This fisheye-lens technique allows users to focus their attention on one (or more) levels within the evolutionary hierarchy while retaining the context within the overall visualization.

Also note that a filled sunburst is fully compatible with the angle detail, detail inside and detail outside fisheye-lens techniques described in [10].

An example of a filled sunburst used to visualize data from a Minority Game simulation is shown in Figure 9.

The second new technique proposed in this paper is a *hybrid sunburst/treemap*. This technique attempts to overcome a fundamental problem of both traditional and filled sunburst visualisation,

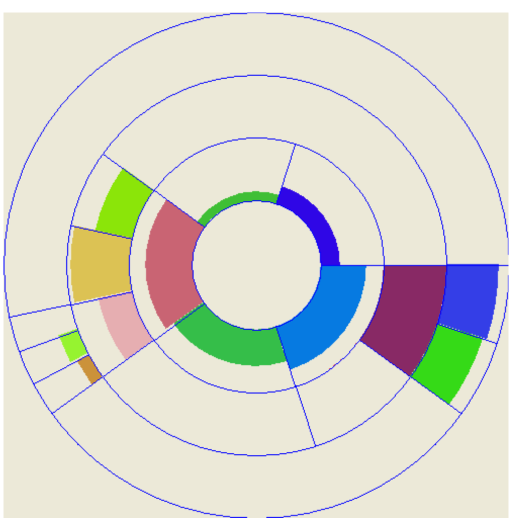
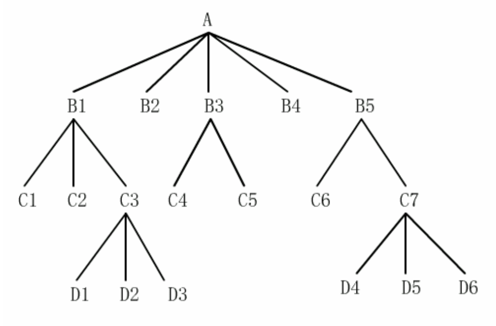


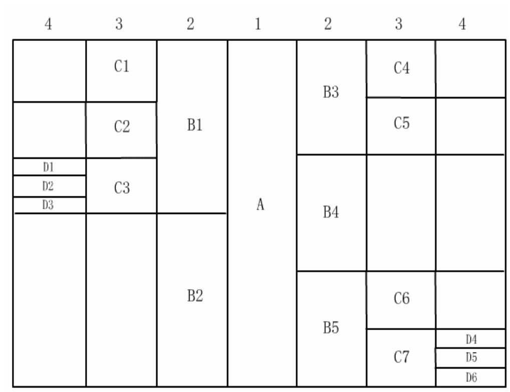
Figure 9: A filled sunburst of Minority Game data

in that they are circular whereas most computer displays are rectangular! The hybrid sunburst/treemap works by placing a rectangle vertically in the centre of the visualisation to represent the root of the evolutionary hierarchy. Then, the children of the root are partitioned so that roughly half are to the left of the root and the rest are to the right. Lower levels of the evolutionary hierarchy are placed beside their parents – child companies are always drawn within the vertical range occupied by their parent companies. A tree data structure and its visualization are shown in Figure 10. The result is somewhat similar to a rectangular cartogram [6]. The same methods of proportionally filling with colour and marking as are used with the filled sunburst visualisation can be used with the hybrid sunburst/treemap visualisation technique.

One disadvantage of the hybrid sunburst/treemap technique is that it still suffers from the "long thin rectangles" experienced by normal treemaps (for example, D1 through D6 in Figure 10). Unlike normal treemaps, squarification cannot be used to delay this problem, as the aspect ratios of the rectangles are largely constrained by the sunburst-like construction of the visualisation. Instead, a



(a) Tree data structure



(b) Hybrid sunburst/treemap visualization

Figure 10: A tree data structure and its hybrid

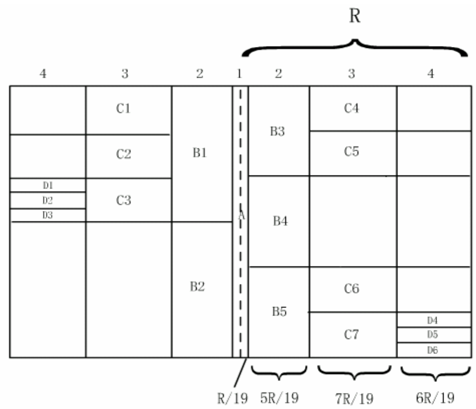


Figure 11: A hybrid sunburst/treemap with DOI applied to width

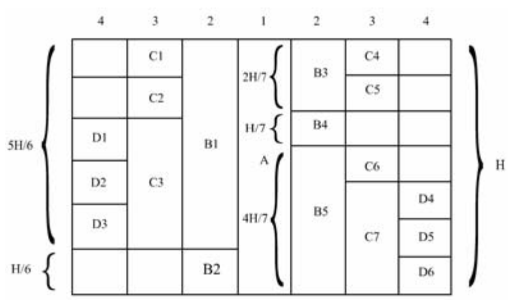


Figure 12: A hybrid sunburst/treemap with DOI applied to height

fisheye-lens technique similar to that proposed by [3] can be used. The width of each rectangle is scaled according to its "degree of interest" (DOI), which is defined as the number of companies in its level of the evolutionary hierarchy divided by the number of companies in total. The result of applying this technique to the visualisation in Figure 10 can be seen in Figure 11. Note that the left and right rectangles at each level have the same width to maintain the balance of the visualisation structure.

The DOI method can also be applied to the heights of subtrees in the evolutionary hierarchy by defining the DOI of each subtree to be the number of companies at the bottom of the evolutionary hierarchy (leaf-companies) in the subtree, divided by the total number of leaf-companies on that side (left or right) of the visualisation. Subtrees are assigned to sides such that the number of leaf-companies on each side is balanced. The result of applying this technique to the visualisation in Figure 10 can be seen in Figure 12.

These DOI methods can be applied to the width and height of the hybrid sunburst/treemap visualisation simultaneously. What is more, the same methods can be applied to the filled sunburst visualisation – level DOI can determine the radius of each circle while subtree DOI can determine the angle occupied by each segment (assuming all subtrees are on the same "side" of the sunburst circle).

An example of a hybrid sunburst/treemap used to visualize the same Minority Game data as Figure 9 is shown in Figure 13. Finally, a filled sunburst with angle DOI visualising the same data is shown in Figure 14.

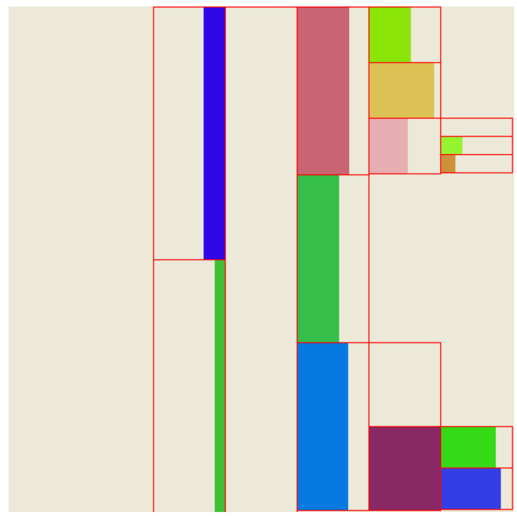


Figure 13: A hybrid treemap/sunburst visualization of Minority Game data

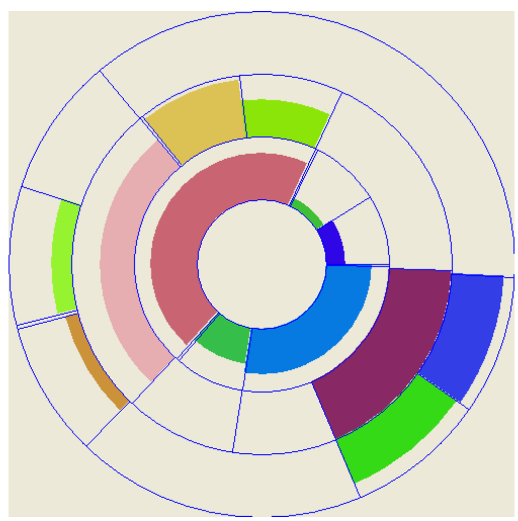


Figure 14: A filled sunburst of Minority Game data with angle DOI

5. CONCLUSION AND FUTURE WORK

In this paper we considered the issue of how users may make the most from large quantities of data produced by many economic models. These users may not be the model developers and may not have an intimate knowledge of the model, yet they may need to quickly understand the implications of the model through its data output. Traditionally model output is presented using the techniques of plots and tables. This paper focused on information visualization techniques.

In this paper, we have presented two new space-filling visualization techniques designed to overcome issues with existing space-filling techniques that were identified by user experimentation. The next step in this research is to evaluate these new techniques through further usability experimentation and compare them against those techniques tested in [4], to evaluate if users can understand the model output more effectively with the new techniques..

REFERENCES:

[1] Challet, D, "The Minority Game's Web Page", 8 Sep 2005. http://www.unifr.ch/econophysics/minority/ [accessed 14 Mar 2006]

[2] Challet, D and Zhang, YC, "Emergence of Cooperation and Organization in an Evolutionary Game", *Physica A*, vol. 246, pp. 407–418, 1997.

[3] Furnas, GW, "Generalized Fisheye Views", *Proc. ACM Conf. Human Factors in Computing Systems (CHI '86)*, pp. 16-23, 1986.

[4] Herbert, RD, Webber, R and Jiang, W, "Measurement of Space-filling Techniques in Simulating the Minority Game", in Zhao, M, and Shi, Z (eds.), *Proc. Int. Conf. Neural Networks and Brain*, Oct 13-15., 2005, Beijing, China, pp. 2021–2027.

[5] Jiang, W, Webber, R, and Herbert, RD, "Information Visualization of an Agent-Based Financial System Model", *Proc. 11th Int. Conf. Computing in Economics and Finance (CEF)*, Washington DC, USA, June 2005. http://gemini.econ.umd.edu/cgi-bin/conference/download.cgi?db_name=SCE2005&paper_id=468 [accessed 14 Mar 2006]

[6] van Kreveld, M, and Speckmann, B, "On Rectangular Cartograms", in *Proc. 12th European Symp. Algorithms (ESA)*, Springer-Verlag, *Lecture Notes in Computer Science*, vol. 3221, pp. 724–735, 2004.

[7] Shneiderman, B,"Tree Visualization with Tree-maps: A 2d Space Filling Approach", *ACM Trans. Graphics*, vol. 11, no. 1, pp. 92–99, 1990.

[8] Spence, R, *Information Visualization*, Addison-Wesley, 2001.

[9] Stasko, J, "SunBurst Page", GVU Center, Georgia Institute of Technology. http://www.gvu.gatech.edu/ii/sunburst/ [accessed 14 March 2006]

[10] Stasko, J and Zhang, E, "Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualization", *Proc. IEEE Symp. Information Visualization*, 9–10 Oct, Salt Lake City, pp. 57–65, 2000.

[11] van Wijk, JJ and van de Wetering, H, "Cushion Treemaps – Visualization of Hierarchical Information" in Wills, G and Keim, D (eds.), *Proc. IEEE Symp. Information Visualization*, pp. 73–78, 1999.

[12] "Treemap: Home Page", HCIL, University of Maryland, 5 Aug 2003. http://www.cs.umd.edu/hcil/treemap/ [accessed 14 March 2006]