FutureQuest. A Sustainability Database Explorer.

Aaron Barsky*

Abstract

The Quest project predicts future living conditions in a region based on policy choices made by regional planners. Each possible future is modeled with 300 measures such as carbon monoxide concentrations, population over 60 years of age, or land allocated to farming. In this paper I introduce FutureQuest, a tool for policy planners to explore a database of possible futures. Presenting a simple interface, FutureQuest requires no training to explore the database. By displaying the results in a visual manner, it is easy to understand the hundreds of measures that describe a future. With a permanent onscreen history of explored scenarios it is easy to navigate through the database. FutureQuest provides a simple interface to explore a large collection of highly varied data.

1 Introduction

Imagine your city 40 years in the future. What would it would it be like? How would you share your vision of the future with someone else? How succinctly could you describe a city?

The Sustainable Development Research Initiative at the University of British Columbia has created models that predict the future for the Georgia Basin region. A single prediction has over 300 measures that explain conditions in the region. These measures vary from the amount of coal burned for electricity generation, to the average amount of time spent in a car commuting each day, to the health of the Sockeye salmon fisheries.

The models have 11 free parameters. Each parameter can take the value of 3 different policy choices that can be set by government planners at a regional level. All 3^{11} possible parameter choices have been computed by the models and the results have been stored in the Quest database. This paper presents a tool to explore the contents of the database.

There are two primary exploration scenarios. The first is to extract a single scenario from the database and explore it in detail. Each scenario has 300 measures. Though the measures have been normalized to a common scale, the units of measurement are for the most part independent. Thus, it is a challenge to give summary statistics. As an example, it would be hard to combine the measures minutes spent in a car and traffic accidents per day into a summary transportation measure as the two scales have nothing in common. The visualization must therefore make it easy to navigate amongst the hundreds of different measures to find areas of interest to the user.

The second exploration scenario is to compare several possible futures to assist in policy decision making. The visualization will present different scenarios on screen at the same time. This allows direct visual comparison which reduces mental effort and eliminates the need to store previous scenarios in the user's mental buffer.

In this paper I will present FutureQuest, a tool for exploring this database of scenarios. This tool will typically be used as part of a one day workshop session. Attendees will typically be regional planners or members of the community with an interest in planning. A session with FutureQuest will last from 5 minutes to an hour depending on the participant's interest in using the software. FutureQuest is thus designed with a very simple user interface. A primary focus of FutureQuest was to make the software attractive and engaging to encourage users to explore scenarios in detail. A goal of a workshop session is to change the behaviour and policy decisions of the participants. During a session with FutureQuest, the policy choices will be constantly displayed while a future is being explored to remind users of their choices.

The remainder of the paper is structured as follows. In section 2 I will discuss related work. Section 3 will present an overview of FutureQuest and its features. Section 4 will present results. Section 5 will discuss the strengths and weaknesses of my solution. Section 6 will discuss the future directions this work could take.

2 Related Work

Creating an interface to a database could be one of the most common software tasks in the whole domain of computing. FutureQuest naturally borrows from several established designs. FutureQuest is not a novel or generic database interaction technique. Instead it is an exposition of good information visualization design practises. The following works have heavily influenced the design of FutureQuest.

QuestVis [8] is a program that visualizes the same database as FutureQuest. It differs from FutureQuest in several important ways. The labels for the measures in QuestVis are not permanently displayed on the screen. Instead, they are only transiently visible while the mouse hovers over an output. This makes it hard to quickly scan outputs and results in less interest from the user. QuestVis similarly allows summary data to be compared between scenarios. However, QuestVis only allows the details of one scenario to be seen at a time. This makes comparison of individual details hard. Finally QuestVis suffered from poor visual appearance. While the data presentation was more accurate than previous attempts at visualizing the data, the inferior aesthetics reduced users' desire to examine the data. The presentation of data values as gradient colour scale boxes in FutureQuest is a direct descendent of the QuestVis project.

Polaris [4] is a generic database visualization tool. It has a visual interface both to design a query and to present the results of the query. It identifies the data in the database as nominative, ordinal, or quantitative. Based on this identification it choose from a variety of data presentation techniques appropriate for the data type. One of the methods Polaris uses to display quantitative data is a diverging colour gradient. This method is used to display the normalized data in FutureQuest.

A Parallel Coordinates Interface for Volume Exploration [3] proposes the idea of using parallel coordinates to link input parameters to the outputs produced. This idea is used as both the query builder in FutureQuest and to link policy choices to the output scenarios.

Using Multilevel Call Matrices in Large Software Projects [5]partitions the screen into a grid. Each cell in the grid is coloured to indicate a feature of the software call structure. A similar grid structure is used to present the scenario details in FutureQuest.

ColorBrewer [1] is a tool to select harmonious colour schemes for maps. This colour chooser greatly enhanced the overall visual appearance of FutureQuest.

*e-mail: barskya@cs.ubc.ca

3 FutureQuest

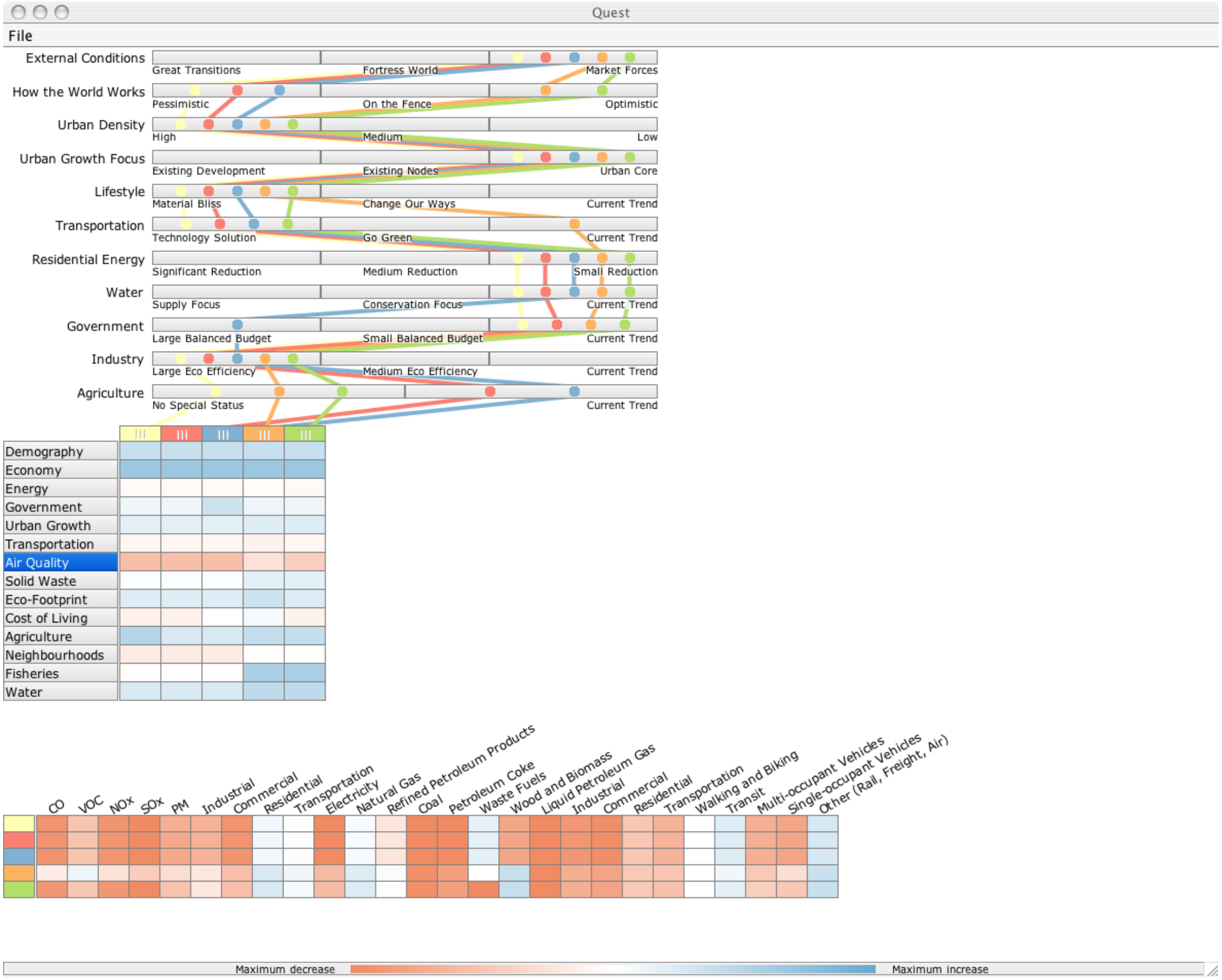

Figure 1: FutureQuest with 5 scenarios being examined

Figure 1 shows FutureQuest with all relevant features displayed on screen. The overall layout follows the information visualisation mantra of overview first, select and filter, details on demand. The top third of the screen shows the visual query builder which selects a scenario from the database. The middle third shows the overview of each scenario in each of the 14 major categories. By clicking on a category we drill down to the measures that belong to that category. Finally the bottom third shows the labeled measures of the category. Each of these sections is now described in detail.

3.1 Screen Parts

3.1.1 Parallel Coordinates as a Permanent Input History

A query is formed by simply choosing from the multiple choice options in each category. In traditional user interface design multiple choice options are represented by radio buttons. These provide only 1 level of history. The previous choice is displayed on screen but earlier choices are unavailable without an external history mechanism. As a primary goal of this software is to link policy choices to the futures that are predicted, it is important to have a full history of policy choices visible on screen. Instead of traditional radio buttons, I chose to use parallel coordinate axes to provide a permanent history mechanism. See Figure 2.

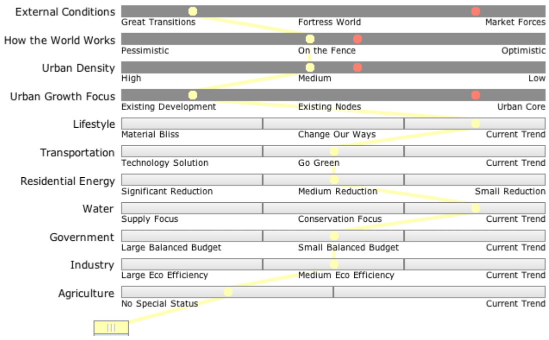

Figure 2: Buttons arranged as parallel coordinate axes

Parallel coordinates are normally used to represent points in continuous multidimensional space. Unlike Cartesian coordinates, where each new axis is placed orthogonally to all other axes, each axis is placed parallel to all other axes [7]. By discretizing our axes into regions representing mutually exclusive input choices, we lose many of the consistent mathematical properties of a parallel coordinate space. However, if we consider a query to be a point in multidimensional space, our discretized parallel coordinates retain their ability to visualize many high dimensional points at once.

The parallel coordinate axis is made up of buttons placed side by side. After a choice is made, a coloured dot is placed on the button. Choices are linked up with lines connecting each of the axes. Finally, the line leaves the final axis and joins to the output. This firmly links the input choices to the produced output. Lines create the strongest Gestalt groupings [6] and the common colour is a secondary Gestalt reinforcement.

It is challenging to use colours to represent different nominal values. As colour perception varies greatly from person to person only a small set of colour can be reliably perceived as distinct. FutureQuest uses a palette of 11 unique colours. After these colours are exhausted, the colours are reused in order. This is unlikely to cause confusion for two reasons. First the lines connecting the dots create strong groupings. Secondly, the dots are displayed in order on the buttons, so there will always be 10 colours separating repeats.

Normally radio buttons are used for mutually exclusive choices. When a new button is pressed, all other buttons pop out and return to their off state easily demonstrating to the user that the choices are mutually exclusive. With my parallel coordinates input, feedback is not so easy. As there are historical entries displayed as dots on the buttons, the user may not notice that her dot has disappeared when a different entry is chosen. To prevent this problem, an axis is disabled after a choice is made. The axis is only re-enabled when the program is ready to query for a new scenario. This is admittedly quite a heavy handed approach to the problem. A more elegant solution would be to animate the dot from the old choice to the new button. The one advantage of disabling the axis is that it hints to the fact that a choice must be made on all axes before a new scenario can be started.

3.1.2 Overview Panel

Each scenario is summarized by a column of values representing the normalized average of the measures in each subcategory. See Figure 3.

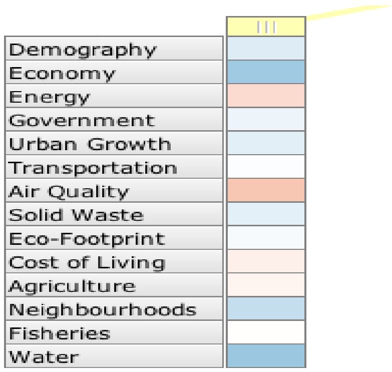

Figure 3: A scenario overview

The values are represented by a colour chosen from a continuous gradient. See figure 4

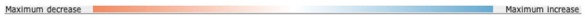

Figure 4: The diverging colour scale

Using colour saturation to represent a value makes it difficult to make precise numeric comparisons. This is in fact what is desired. The models represent predictions for 40 years in the future and will necessarily be inaccurate. We wish to communicate the gist of the model to the user. We would not want the user to form an opinion that Scenario A is preferable to Scenario B because A has a carbon monoxide concentration of 0.79 where B had a concentration of 0.83.

It is easiest to compare two scenarios when they are physically located beside each other on screen. Thus I have allowed the columns to be rearranged through a simple drag and drop interface. Reordering scenarios also reorders the linked details view. See Figure 5.

Details for a category are viewed by clicking on a category button. Alternately, clicking in a summary box will bring up the corresponding details view. The chosen category stays highlighted to indicate which details are being viewed at the bottom of the screen. See Figure 6.

3.1.3 Details on Demand

Figure 7 shows the details section, located in the bottom third of the screen. This is where the actual database values are presented.

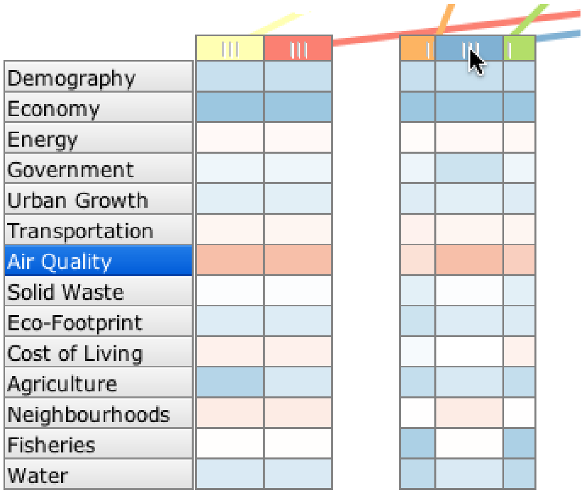

Figure 5: Drag and drop reordering

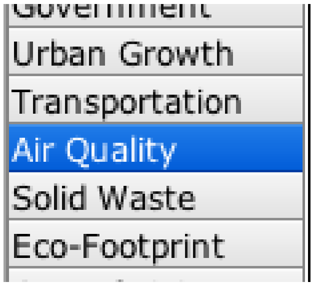

Figure 6: Category selection

All of the values in the database have been normalized to a -1 to 1 scale. Like the overviews, these values are represented as colours chosen from a divergent colour gradient.

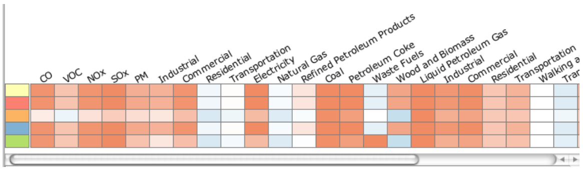

Figure 7: The category details

All of the scenarios chosen so far are presented in rows. The user can identify the input choices that led to these values by matching colours. They are also presented in the same order as the summaries to maintain context.

Labels identifying the measure are placed at the top of the column. I chose to place the text at the head of the columns (rather than arranging the data in rows) to take advantage of the higher horizontal resolution of most monitors. Rotated only 30 degrees from the horizontal the label remain easily readable and yet densely packed.

It is easy to scan the labels for measures of interest and then make comparisons down the columns. One disadvantage of this layout is that it also encourages making comparisons across the rows. Unfortunately, due to the different scales, units, and meaning of each measure, such comparisons are meaningless.

3.2 Graphic Design

Data visualizations are used in two different ways. The first is for the viewer to gain insight into the data by examining it with the visualization. The second is for someone who understands the data to use the visualization as a teaching tool to share the insights with others. When presenting the data in teaching mode, it is vital for aesthetics to rule. The viewer may never have seen the visualization before. Thus the presentation should be kept simple so that the new viewer can understand it. The presentation should be beautiful and engaging so that the viewer wishes to examine it in more detail and spend the time to understand it. A dense display full of information may be useful to someone looking for new insights from his data, but it would quickly overwhelm a new user.

FutureQuest is a presentation and teaching tool. The creators of the future models know how their figures were created and how the values are linked to the input parameters. FutureQuest is not for the creators. It is instead a way for the creators to share their data, which they know intimately, with the general public. I have therefore spent a high degree of effort in making the screens of FutureQuest pleasing to the eye.

3.2.1 The Subtleties of an Attractive Design

Though it was implemented in Java, FutureQuest does not have the distinct Java appearance. All of the controls implement custom painting routines to give the application a beautiful and harmonious appearance. Buttons have slight colour gradients lending to a subtle three dimensional feel. The colours that label scenario choices all have the same saturation. Grids are painted with a lightly coloured thin single pixel border. This emphasizes the grid content instead of the grid itself and prevents visual artifacts from appearing at grid intersections. Similarly the category buttons are slightly overlapping to create thin light borders. The connecting lines in the parallel coordinates interface are antialiased for a smooth even appearance. A semi-opaque mask is rendered under labels so that the line continuations can be followed without obscuring the label text.

All text is rendered using Verdana. This font has generous spacing between letters, plenty of internal room in the loops of letters, and no serifs that would look pixelated when read onscreen. This font is legible even where the text is rotated in the details section. Due to rounding errors when rotating text, many other fonts are an illegible mess in any orientation other than horizontal.

4 Results

All the results that follow come only from my personal use of FutureQuest. Clearly a user study is required to validate any of these results.

4.1 Input History

Using parallel coordinates as a permanent input history led to limited success. The visual memory of the dots on the buttons helps the user to avoid repeating scenarios. The dots make it easy to create a new scenario that differs in only one input dimension. Organizing the summaries to cluster in the same way as the choices for a particular axis makes it easy to see what effect the axis choice has. See Figure 8

The connecting lines between axes are useful up to about 5 scenarios. The lines enhance differences in inputs and crossovers show where input decisions differ. When the mouse is brushed over a scenario, its input lines are thickened and its input lines are raised to the top so that they cross over all other lines. Unfortunately this does not scale well to larger numbers of scenarios. The parallel coordinates in Figure 9 shows the chaos that arises when there are too many lines. I increased the brushing to also highlight the selected button labels as an additional help to sort through the clutter.

I was at first tempted to eliminate the connecting lines altogether. After all, much of the benefit of the parallel coordinates input is realised by the dots on the buttons which scale well. However, the lines are useful in small doses, and they also serve the vital function of grouping inputs once the colour choices are recycled.

I would propose the following scheme for the future. Inputs would be ranked with a degree of interest. Newly created inputs or clicked inputs are raised to top rank. Hovering a mouse over an input would slowly accumulate rank by fractional amounts. Most likely this accumulation would be non-linear. The top four lines would be rendered in full colour. The next 7 lines would be rendered slightly thinner and with increasingly faded colour. Finally all remaining lines would be rendered in a single pixel width in grey only.

As the project progressed, I was able to devote less and less screen space to the parallel coordinates input as space was taken up by the overview and details. As the space between axes reduced, so did the value of the lines connecting them. Lines nearing horizontal, such as those that connect a button 1 to a button 3, provide the least useful feedback. This situation could be improved in a few ways. I could squash the labels under the buttons and shrink the buttons horizontally. This would improve the horizontal to vertical aspect ratio and cause connecting lines to be drawn more vertically. The buttons could be shrunk vertically giving more room to actually see the connecting lines. This would unfortunately have the side effect of making the buttons more difficult to click. Finally, more screen space could be allotted to the input by a redesign of the screen layout.

4.2 Understanding the Quest Database

4.2.1 Engagement

The attractiveness of the display encouraged me to use FutureQuest. Instead of working at adding features, I often found myself simply clicking new inputs to see more beautiful lines and circles appear on the display. This had the unintentional side effect of causing me to explore many scenarios in the Quest database. While I started with thoughtless clicking, I was soon intrigued by the constant orange boxes in the Air Quality summary. See Figure 9.

I challenged myself to make Air Quality turn blue. I continued for a short while with thoughtless clicking, before realising that I was unlikely to randomly find a blue box in 3^{11} possible scenarios. I then changed my mindset and tried to pick the policy choices that would result in the best air quality. FutureQuest succeeded in causing me to consider policy choices for the future. Sadly, in the course of this exploration I discovered a fatal flaw in my understanding of the Quest database. I will discuss this further in 4.2.3

4.2.2 Finding 0 Variance Data

Low variance data is very easy to spot when the data is presented as a colour gradient. Soon after running FutureQuest it became clear that the Economy box never substantially changed value. Drilling down for details, we find columns of homogenous colour. Figure 10 shows us that only 5 of the 21 measures ever change value. An SQL query for unique values in the database verified this suspicion. As they have 0 variance we might suspect that these measures are not being properly set by the future models. A discussion with the model designers will be required to investigate this possibility.

4.2.3 Understanding the Unique Measure Scales

FutureQuest led me to thoroughly understand the contents of the Quest database. In fact, it led me to an insight whereby I realised I had poorly designed the visualization based on my previous understanding of the data.

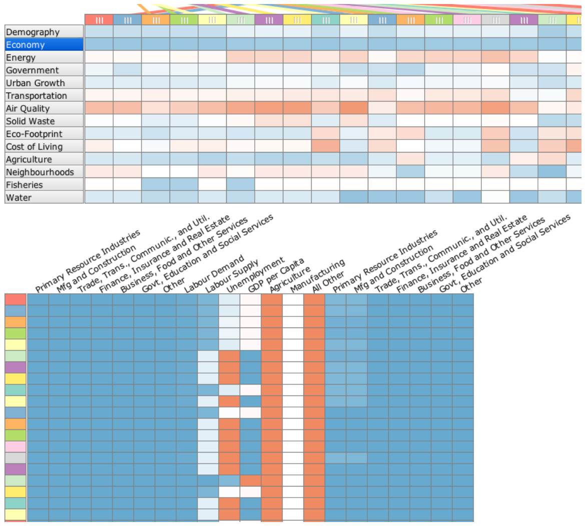
Demography
Economy
Energy
Government
Urban Growth
Transportation
Air Quality
Solid Waste
Eco-Footprint
Cost of Living
Agriculture
Neighbourhoods
Fisheries
Water
Primary Resource Industries
Mfg and Construction
Trade, Trans., Communic., and Util.
Finance, Insurance and Real Estate
Business, Food and Other Services
Govt, Education and Social Services
Other
Labour Demand
Labour Supply
Unemployment
GDP per Capita
Agriculture
Manufacturing
All Other
Primary Resource Industries
Mfg and Construction
Trade, Trans., Communic., and Util.
Finance, Insurance and Real Estate
Business, Food and Other Services
Govt, Education and Social Services
Other

Figure 10: The economy never changes

While involved in a FutureQuest session, I had challenged myself to turn the Air Quality summary blue. I chose policies that I felt sure would lead to cleaner air, but the summary stubbornly stayed orange. Examining the details of Air Quality, I soon concluded that the summary square is meaningless. My ideal world wanted an increase in Transit and Walking and Biking, but a decrease in Carbon Monoxide and Coal burning. Not only were the scales wildly different (% of trips by transit vs. parts per million of carbon monoxide vs. tons of coal used for electricity generation) but even the sign of values were different. My personal notion of a healthy future wanted positive values for Transit and negative values for Carbon Monoxide. Did I want the Air Quality summary to be blue, orange, or white? If it turned blue, were the right subcategories turning blue for my ideal world?

I concluded that the summary columns aren't providing any useful information. At best, they provide a similarity score that could judge how similar two scenarios are. This feeble function does not justify using one third of the screen space. Further I would use a different colour scale for the summaries so that no one glances at Air Quality and judges it increased or decreased.

One solution to this problem would be to change the database values so that positive values represent good results and negative values represent bad results. The scale would be relabeled from most increased/most decreased to best/worst. But it is precisely these kind of value judgements that the future modellers are trying to avoid. Are growing suburbs an increase in affordable housing (good) or nightmarish suburban sprawl (bad)? I would instead conclude that summaries have little value and their screen real-estate should be replaced with more details about the measures.

4.3 Performance

The application was tested on an aging PowerPC G4 1.33Ghz with 512 mb of RAM with a 32GB AGP graphics card at a resolution of 1600x1200. As a simple 2-Dimensional display, there were no noticeable delays painting the screen. This was true even when there were enough scenarios selected to fill the entire screen space.

Problems did occur with network round trip times to fetch data from the database. A query is run for each scenario each time a new details screen is selected. On the local database server, an SQL query for a single scenario's details ran in 10 ms. However, with a full network roundtrip for each query, this drill-down delay could extend into the order of seconds. In the future, database queries will be placed on a separate thread so that the application remains responsive while the data is retrieved from the network.

5 The Good and the Bad, but never Ugly

5.1 Strengths

FutureQuest succeeds in many areas. By emphasizing aesthetics, FutureQuest is an application that I used repeatedly and still want to play with. Presenting the data through FutureQuest I was able to understand the Quest database well enough to want to redesign the presentation of my data.

The permanent history of the parallel coordinates interface promoted exploration of the database. A blank button would call out to be pressed to find out what its effect might be. The history makes it easy to vary choices, or keep choices similar, depending on user desires.

Presenting the details as coloured boxes with column labels, it is easy to see an entire category of measures at once and find measures of interest.

5.2 Lessons Learned

The largest fault is that the summary columns provide useless, and perhaps even misleading, information. Clearly these columns should not occupy one third of the screen.

I quickly discovered that text takes up a large amount of screen real-estate, regardless of screen resolution. Even on the modern 2048x1600 resolution display, there was no extra space as the font sizes needed to be increased to keep text legible. In the future, I will count the rows of text needed in my prototypes. If they exceed 30, I will immediately redesign.

Lacking sufficient screen space, I rotated text in the vertical column headers. While this is an effective solution, it is peculiar that the summaries are presented in columns and the details are presented in rows. This causes extra mental effort to interpret the data.

I highlighted the category button to show which row was being expanded for drill-down. Once my gaze scanned more than 10 columns to the right of the button I lost track of which row I was examining. The drill-down row could have been more effectively highlighted by adding white-space above and below it.

The parallel coordinates interface did not scale well to long sessions. I have already outlined a proposal for improving its scalability.

6 Future Work

6.1 Input Parallel Coordinates

I often wanted to change my choice on an input axis for two different reasons. The first is that I made a mistake, and simply wanted to move my choice. The second is that I wanted to create a new scenario that was the same in all axes except one. I can think of two models to accommodate these two actions. The fist would be to allow changes and have a button to manually create a duplicate scenario. The second would be to create a new scenario with each input change and then have a command to easily delete unwanted intermediate scenarios.

I found it interesting to reorder my outputs into clusters that corresponded to input choices for a particular axis. I did this often enough to want it as an automatic feature.

I have already outlined one idea to improve the scalability of the parallel coordinates. Another more challenging solution would be to reorganise the dots within the buttons to produce less line crossings. Another approach would be to aggregate lines heading in common directions into a single thick stream. This would certainly require more intermediate space between axes.

6.2 Annotations, Saving, and Playback

After a session with FutureQuest, a policy planner might like to share her insights to justify her policy making decisions. FutureQuest has a basic load and save feature. It would be better if we could save the entire session. The user could then playback the session and select key screens for display. I would like the ability to name and label favourite scenarios. I would also allow annotations anywhere on the screen that would be displayed during playback.

6.3 Explaining the Measures

Currently measures are only explained as a label and a value. This provides a great overview, but at some point, the user will be interested in what the measure actually represents. FutureQuest needs one more level of drill-down. When clicking on a measure information such as the scale, units, current value, and future value should be shown. If available, help text that adds context or meaning to the measure should also be displayed.

7 Conclusion

I have presented FutureQuest, a system for exploring an environmental sustainability database. The use of parallel coordinates as a combination input and history device was successful at guiding exploration through the database. The coloured columns of details were able to present an entire category of measures to the user at once. It was easy to compare details from different scenarios. The presentation of summary data was found to be inaccurate and misleading and will be removed from future versions. By focusing on aesthetics, I have created an application that is engaging and easy to use.

References

[1] Cynthia A. Brewer. Color brewer. http://www.ColorBrewer.org.

[2] Kitware, Inc. *The Visualization Toolkit User's Guide*, January 2003.

[3] Torsten Moller Melanie Tory, Simeon Potts. A parallel coordinates style interface for exploratory volume visualization. In *IEEE Transactions on Visualization and Computer Graphics*, volume 11, pages 71–80, January 2005.

[4] Chris Stolte, Diane Tang, and Pat Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 112–122, New York, NY, USA, 2002. ACM Press.

[5] Frank van Ham. Using multilevel call matrices in large software projects. In *IEEE Symposium on Information Visualization*, volume 11, page 29, 2003.

[6] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, second edition, 2004.

[7] Edward Wegman. Hyperdimensional data analysis using parallel coordinates. In *Journal of the American Statistical Association*, pages 664–675, 1985.

[8] Matt Williams. Questvis and mdsteer: The visualization of high-dimensional environmental sustainability data. Master's thesis, The University of British Columbia, 2004.

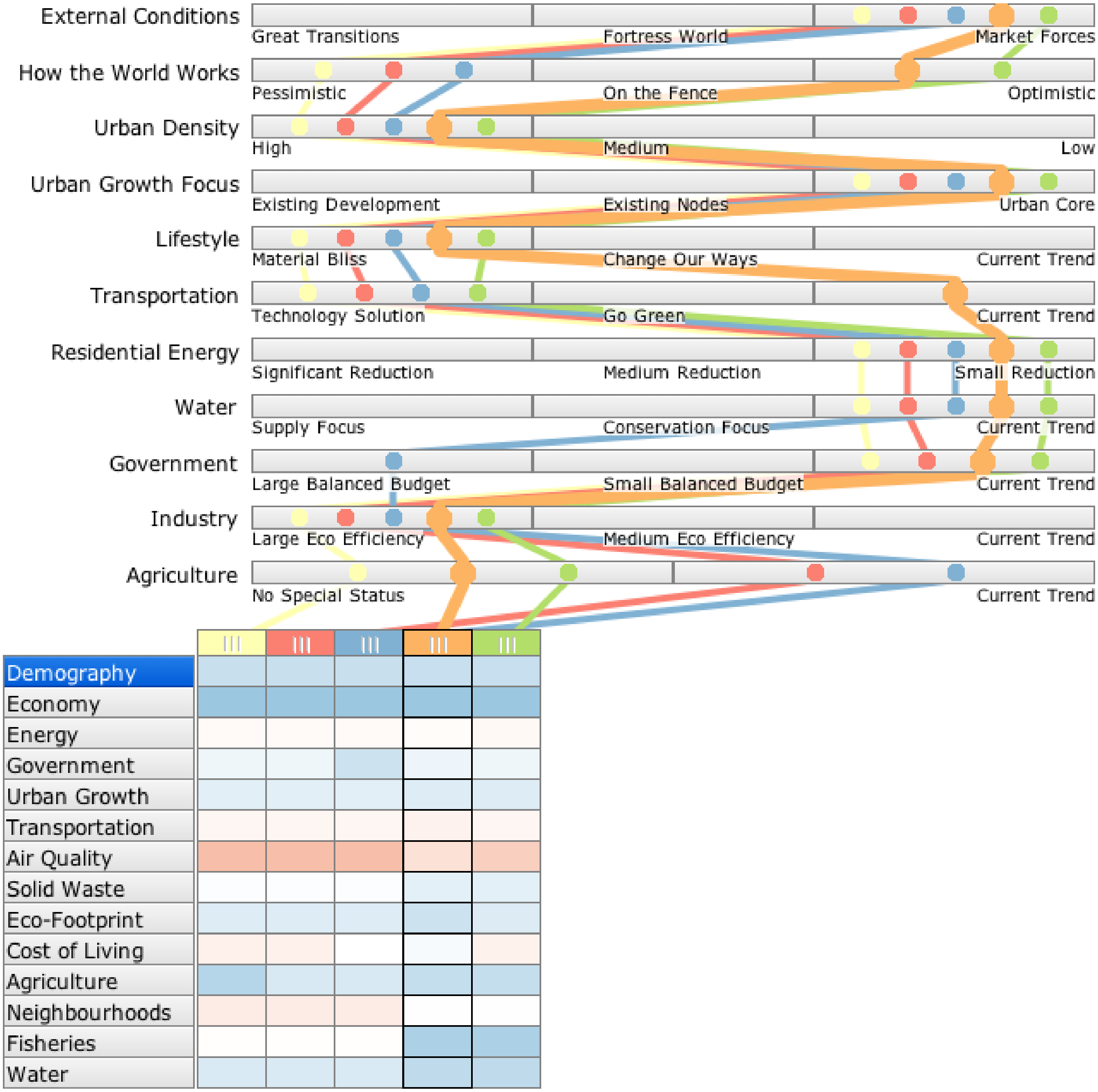

Figure 8: A successful use of parallel coordinates as input. In this example, the summaries are arranged in the same order as the How the World Works axis to help understand the effect of this policy.

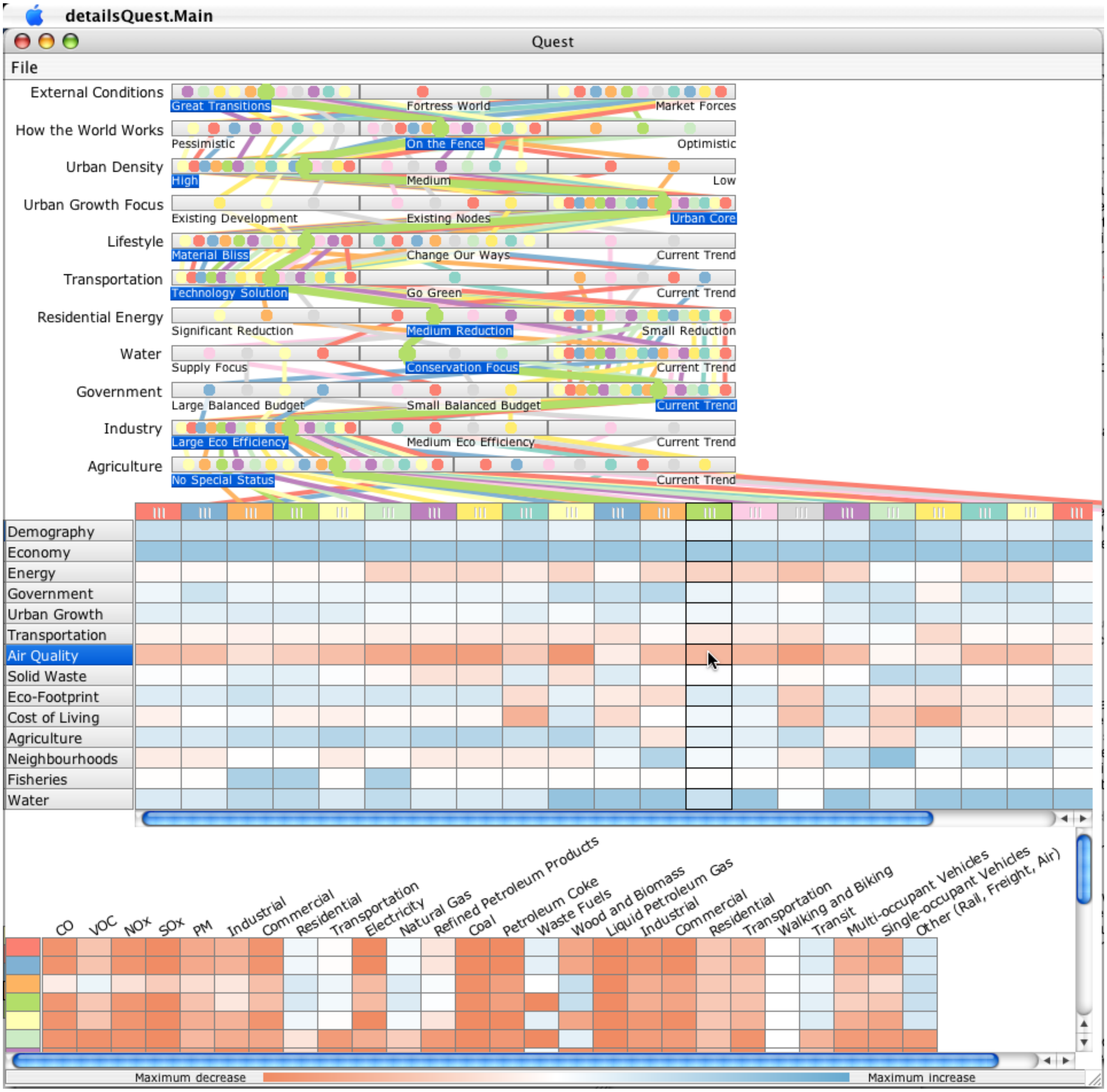

Figure 9: Why is Air Quality always orange?