Reflections on QuestVis: A Visualization System for an Environmental Sustainability Model

Tamara Munzner, Aaron Barsky, and Matt Williams
University of British Columbia

Abstract

We present lessons learned from the iterative design of QuestVis, a visualization interface for the QUEST environmental sustainability model. The QUEST model predicts the effects of policy choices in the present using scenarios of future outcomes that consist of several hundred indicators. QuestVis treats this information as a high-dimensional dataset, and shows the relationship between input choices and output indicators using linked views and a compact multilevel browser for indicator values. A first prototype also featured an overview of the space of all possible scenarios based on dimensionality reduction, but this representation was deemed to be be inappropriate for a target audience of people unfamiliar with data analysis. A second prototype with a considerably simplified and streamlined interface was created that supported comparison between multiple scenarios using a flexible approach to aggregation. However, QuestVis was not deployed because of a mismatch between the design goals of the project and the true needs of the target user community, who did not need to carry out detailed analysis of the high-dimensional dataset. We discuss this breakdown in the context of a nested model for visualization design and evaluation.

Keywords: high-dimensional visualization, dimensionality reduction, linked views, simulation visualization

1 Introduction

Most people have a vision of the future they desire. A desired future might have low unemployment, or less traffic, or clean air, or maybe all of the above. However, people are often unaware of how the interplay between regional policy choices made in the present could either bring about or prevent desired aspects of these futures.

The Georgia Basins Futures Project created an interactive software system called QUEST (Quite Useful Ecosystem Scenario Tool) to engage the public in dialogue about sustainability choices and outcomes [4]. QUEST contains computational models that generate future scenarios by predicting living conditions over the next forty years for the Georgia Basin region. These models are informed by expert understanding of ecological, social and economic systems. They have free parameters that users can set to reflect assumptions about how the future will unfold, governmental policies that will be implemented at a local level, and individual behaviors. QUEST is designed for use by lay people interested in learning about sustainability issues in a facilitated workshop format lasting less than a single day. The goal of the project is to increase a participant's understanding of how current decisions affect the sustainability for the region. Its underlying philosophy is that the real purpose of modeling is not to describe the world, but to change the thinking of users. After its initial development at the Sustainable Development Research Institute at the University of British Columbia, QUEST has been further refined by the spinoff company Envision Sustainability Tools, Inc.

A single QUEST prediction has 294 indicators that characterize a future scenario. Example indicators include the amount of coal burned for electricity generation, the average amount of time spent in a car commuting each day, and the health of the sockeye salmon fisheries. Any change to a model parameter will generates a new scenario, where some or all of these 294 indicators may have changed.

From an information visualization perspective, we can consider these QUEST predictions as a high-dimensional data set with 294 dimensions. The goal of the QuestVis project was to use information visualization techniques to allow individual users to explore this rich data set of scenarios more extensively, both within and between scenarios. In this paper, we will present the design of the two interactive prototypes that we iteratively developed. The main contribution of this paper is the reflection on the lessons learned in this process.

We begin in Section 2 with background about the usage of the current QUEST system in workshops. Section 3 presents the design choices of the first prototype, and Section 4 discusses its limitations. We continue with the design of the second prototype in Section 5, and then reflections on its strengths and limitations in Section 6. We then discuss the related work in Section 7, and conclude in Section 8.

2 QUEST Usage

The QUEST software was designed for use in a large-group workshop format with a trained facilitator, rather than for individual exploratory use. A QUEST session has three sequential stages: an input stage where the group makes policy decisions by specifying the choices for a scenario, a model computation stage to which generates scenario indicators as output, and an analysis stage where future scenarios are understood through inspection of these indicators. Optimizing for some indicators may force others to have less desirable values. QUEST is intended to help lay people understand these tradeoffs at a deeper level than they did before participating in a workshop. These people are often citizens in a particular community that are wrestling with thorny policy choices, and QUEST workshops are used as part of a public engagement process.

In the input stage, up to 49 policy choices can be made. For example, the waste reduction policy choice controls which of five levels the scenario will use, from a maximum of `significant reduction` to a minimum of `same as now`. The facilitator leads the group through these choices, with significant discussion of possible consequences of their decision making. These choices are grouped into categories, and QUEST supports "presets" that can specify many input choices at once within a category. For example, Figure 1 shows that the `Urban Development` options has four presets: `Current Trend`; `Protect Air, Water and Land`; `Strengthen Communities`; and `Live and Let Live`. Choosing one of these presets sets all seven of the input choices, although any of them can be explicitly overridden after that choice is made. At a higher level, there is a two-phase structure, with a division into the `Set Context` and `Choose Options` phases.

After all input decisions are made, the facilitator explicitly starts the model computation phase, which takes approximately two minutes. That time is typically spent in continued group discussion.

The third stage of QUEST usage is the analysis phase. Figure 2 shows the overview screen for the scenario that presents 9 of the indicators in radial chart form, with the values for today overlaid with the values forty years from now as a radar graph. The 294 output indicators are categorized into a two-level hierarchy with a total of 88 detailed views featuring bar charts, which can be selected by drilling down through the `Scenario Navigation` result menus. Figure 3 shows two of these views, which are used to understand specific aspects of the chosen future scenario. The input variables that could affect the indicators in question are shown on the right, but they cannot be changed during this analysis stage.

The entire facilitated process of choosing the input decisions, computing the future, and analyzing a single chosen future scenario typically takes over an hour. The facilitator does not typically lead the group through an exhaustive analysis of every single indicator, but focuses on the most relevant ones. In a workshop, this process is typically repeated a few times to analyze a small number of scenarios.

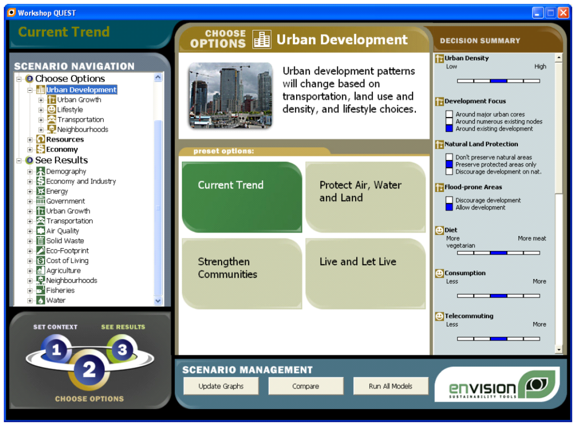

Urban development patterns will change based on transportation, land use and density, and lifestyle choices.

Figure 1: Making policy decisions in the QUEST input stage. The `Scenario Navigation` menu on the left presents categories of options. One of the green `preset options` in the middle can be chosen to specify an entire set of choices at once, or the user can make detailed choices for individual options on the right.

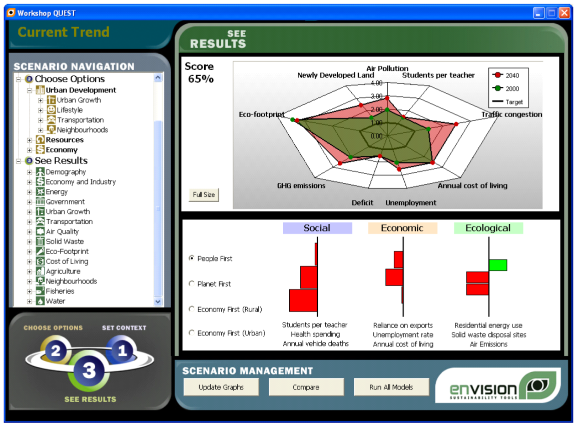
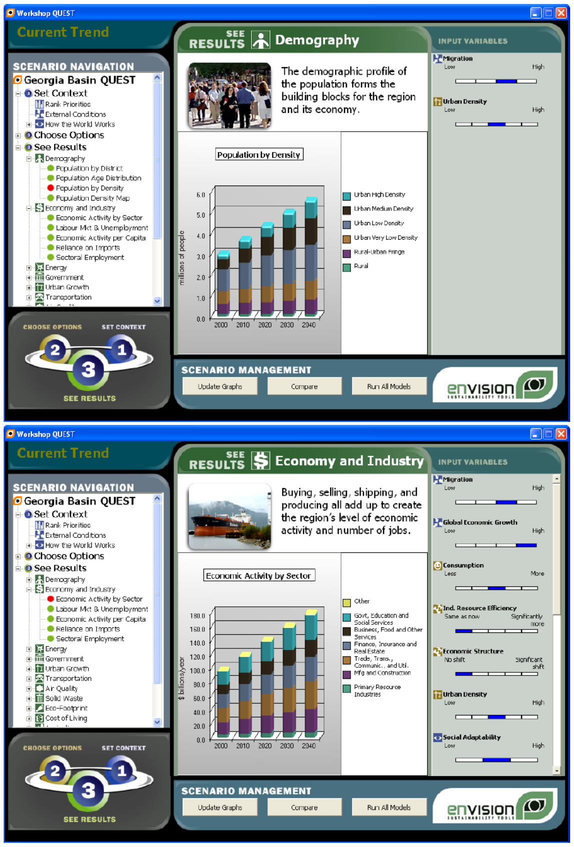


Figure 2: The QUEST analysis stage begins with the overview page, showing a carefully chosen subset of 9 of the 294 output indicators using a radial chart.

2.1 QUEST Strengths and Limitations

Many design decisions in QUEST were made with careful attention to the difficult nuances of how to use technology to engage citizens in an informed sustainability discussion, with the ultimate goal of not only furthering their understanding but changing their behavior. The user interface has a very polished graphic design, a very appropriate choice for its intended audience of the nontechnical lay public. The software is well suited for a detailed investigation of a few key indicators, as part of an in-depth facilitated group discussion.


The demographic profile of the population forms the building blocks for the region and its economy.

Buying, selling, shipping, and producing all add up to create the region's level of economic activity and number of jobs.

Figure 3: Detailed analysis of the 294 QUEST output indicators for the chosen scenario, by picking one of the 88 possible windows from the hierarchical `Scenario Navigation` menu on the left. **Top:** The `Population by Density` view shows six of the indicators as stacked bar charts. **Bottom:** The `Economic Activity by Sector` view shows seven other indicators. The input choices related to each view are shown on the right as read-only displays that cannot be changed at this stage.

However, when considering the capabilities of the software as an interface that could allow the exploration of a rich information space, we identified a significant limitation. The tool does not provide support for synthesizing a high-level understanding of how specific input choices allow or preclude whole sets of scenarios. The reason for this limitation is two-fold.

First, it is difficult to compare indicators from different categories within a single scenario because they are presented on separate screens. Individual screens have a low *information density*, using a large number of pixels to show only a few variables. To comprehend the relationships between the highly interdependent output indicators, for example whether an attempt to improve water quality will always result in an economic loss as measured across several possible indicators, people must internally synthesize results from multiple screens. However, research has shown strict limits on the capacity for people to conduct a detailed comparison of remembered images with their current view [12, 17]. Moreover, the overview only presents information about nine specific indicators. If the user is interested in any of the 285 other indicators, the overview screen is not useful.

Second, comparing indicator values between different scenarios is even more difficult, so it is difficult to make a judgement about the quality of a particular scenario compared to other possible alternatives. The central problem is the three-stage architecture where interaction is separated into two distinct phases, setting input choices versus analyzing output indicators, with a wait of multiple minutes between the two. A fundamental theory of interaction design separates human cognitive response to systems based on response time, with major differences at three threshold limits: multisecond, one second, and subsecond wait times [3]. The multi-minute gap precludes any kind of tight coupling where the result of changing an input choice on the output indicators is immediately visible. QUEST displays the specific inputs that might affect the output indicators on a given screen in the analysis phase, but they are read-only and cannot be changed. Moreover, setting the inputs a relatively cumbersome multi-step operation, because the user must use a menu to navigate to the multiple views to make input choices across many categories.

Finally, the need for a trained facilitator is a major resource constraint. An earlier version of QUEST did try to support individual rather than facilitated group exploration, but it was deemed too difficult for lay users to understand the complexities of the problem without a trained facilitator.

3 First Prototype

We created QuestVis, shown in Figure 4, a fully interactive prototype with an interface designed for interactive exploration of data generated by the existing QUEST computational models. It was designed primarily to support individual exploration, although it could also be used by a facilitator in a group setting. Our hope was that powerful interactive exploration capabilities would allow nontechnical users to directly synthesize an understanding of the linkages between current policy choices and future outcomes, without the need for a facilitator.

We made three major decisions in the design of the first QuestVis prototype. First, we provided linked views of the input choices and the output indicators that were always visible, with the ability to change any input choices at any time and see immediate visual feedback of how the indicators changed in response. Second, we created a multilevel overview of the indicators that was compact enough to see all indicators within a scenario simultaneously when fully expanded, or to compare between scenarios at higher levels of aggregation. Third, we showed where chosen scenarios fell within a visual representation of the space of all possible scenarios.

3.1 Linking Input Choices With Output Indicators

The goal of enabling immediate changes for output indicators in response to input choice changes required us to use a very different system architecture than the original QUEST. We generated a database of precomputed scenarios, removing the need for a model computation stage and for a separation between specifying input and analyzing output stages. The set of input choices is used as the key for the database, which returns the 294 output indicators associated with that particular scenario. The database was generated by running the computational model in batch mode, and saving the computed indicators. This architecture would not be feasible with the full set of 49 input decisions that each had four possible choices, yielding the intractable number of 10^{30} scenarios. The model creators at SDRI and Envision chose the 11 most influential input decisions to use in creating the database. Ten of these inputs had three possible choices, while one input had two choices. The resulting database was a manageable size, containing $3^{10} * 2^1 = 118{,}098$ scenarios.

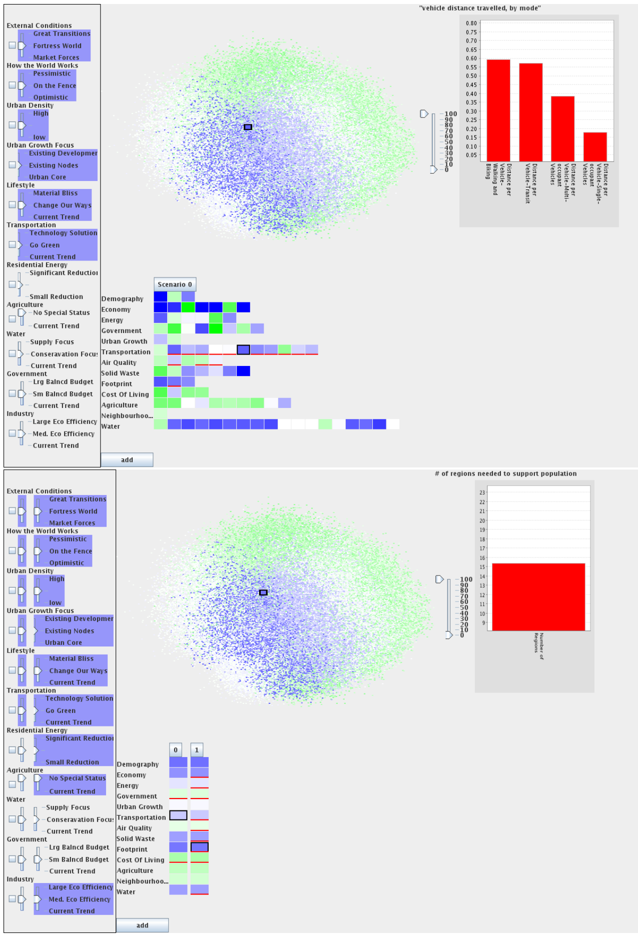
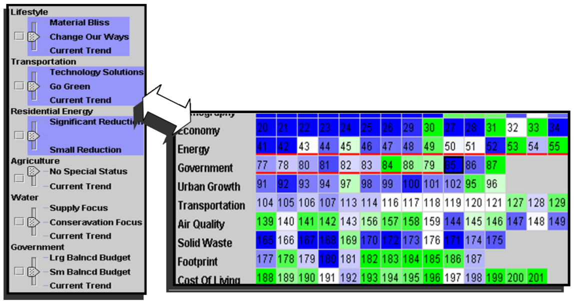


Figure 4: The first QuestVis prototype. **Top:** Exploring a single scenario, with the multilevel indicator expanded to the middle level. **Bottom:** Comparing two scenarios at the highest level of aggregation.

QuestVis shows input choices and output indicators side by side on the same screen. The full set of 11 input choices are displayed in a panel on the left, with sliders that visually indicate the current choices and can be changed at any time. When an input choice is changed, the output indicators immediately update. QuestVis is architected as a front-end Java client that connects to a back-end database running on a separate machine. Subsecond response time for an update is typical when a high-bandwidth connection between these two machines is available.

In addition to real-time responsiveness, we create an explicit visual linkage between the input and output displays using linked highlighting. In addition to the large data set that is the mappings from input to output choices for each scenario, the database also contains a list of which input choices can affect each indicator, and which indicators can change when a particular input choice is changed. The linked mouseover highlighting makes this information visually apparent to the user at all times, as shown in Figure 5. When the user moves the cursor over an output indicator, the input sliders that can affect it are highlighted by changing their background color to light blue. Conversely, when the user moves the cursor over an input slider, the output indicators that could change when it is moved are highlighted with a red underline. Our conjecture was that the combination of real-time response and linked highlighting would allow users to understand the consequences of input choices through direct experimentation and exploration, without the need for guidance from a facilitator to explain the consequences of environmental choices during an extensive and separate input decision phase. In QuestVis, exploration can take place in a tight loop that allows users to see the impact of a few policy changes per second, as opposed to the time scale of a few choices per hour in the previous paradigm.



Economy 20 21 22 23 24 25 26 29 30 27 28 31 32 33 34
Energy 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
Government 77 78 80 81 82 83 84 88 79 85 86 87
Urban Growth 91 92 93 94 97 98 99 100 101 102 95 96
Transportation 104 105 106 107 113 114 116 117 118 119 120 121 127 128 129
Air Quality 139 140 141 142 143 156 157 158 159 144 145 146 147 148 149
Solid Waste 165 166 167 168 169 170 172 173 176 171 174 175
Footprint 177 178 179 180 181 182 183 184 185 186 187
Cost Of Living 188 189 190 191 192 193 194 195 196 197 198 199 200 201

Figure 5: Bidirectional linked highlighting between input choices and output indicators. **Left**: The input choice sliders that can affect an output indicator are highlighted on mouseover with blue backgrounds. **Right**: All output indicators that an input slider can affect are highlighted on mouseover by underlining them in red.

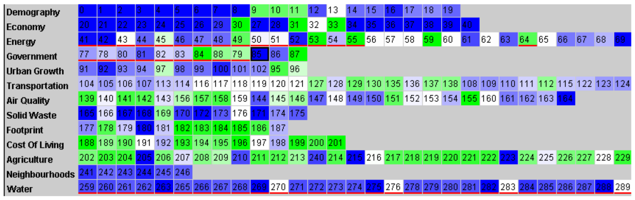
3.2 Multilevel Indicator Browsing

QuestVis provides indicator browsing in two ways. There is a detail view very similar to QUEST, with a dedicated window for showing a small number of indicators in a single bar chart, using spatial position to encode the indicator values. However, considerably less screen area is devoted to this window than in QUEST.

QuestVis also has a large overview region that shows information about indicators at three possible levels. The multilevel indicator browsing is supported using a compact visual encoding of colored boxes, as in the heatmap views that have become popular in bioinformatics [19], rather than the radar graph of Figure 2 that requires a lot of display area because the information is encoded using spatial position.

Each colored box showing an indicator value is several pixels on a side: large enough to avoid difficulties with small-field color perception, but much smaller than would be required to encode the data spatially. We use a diverging color scale, where white represents no change compared to the present-day value. Saturation represents the size of the increase or decrease from this value, with fully saturated blue for the maximum increase and fully saturated green for the maximum decrease. These values are normalized for each indicator relative to the its minimum and maximum values across all scenarios in the database.

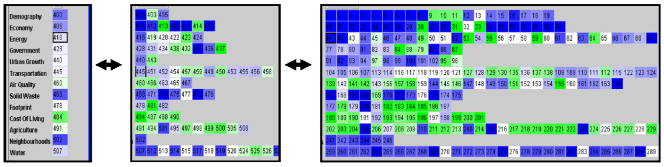
Figure 6 shows a zoomed-in view of this overview region in the fully expanded state, when the full set of 294 output indicators for a scenario are shown simultaneously. The indicators are organized into thirteen categories, such as `Energy`, `Transportation`, `Water`, and `Air Quality`, and each is shown in a separate row. The fully expanded state allows easy comparison between all of the indicators for a scenario. Clicking on an indicator box in the fully expanded view changes the detailed bar chart shown in the upper right corner. The full overview can be collapsed to a mid-level view showing only the most important indicators for each category, or to a high-level view showing only one colored box for each category with an aggregate value that is the average of all indicators in the category, as shown in 7. Clicking on the scenario name at the top of the column triggers the expanding or collapsing behavior. The organization of indicators into categories and the choice of which indicators to use for the mid-level overview was provided by the model creators.



Demography 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Economy 20 21 22 23 24 25 26 29 30 27 28 31 32 33 34 35 36 37 38 39 40
Energy 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
Government 77 78 80 81 82 83 84 88 79 85 86 87
Urban Growth 91 92 93 94 97 98 99 100 101 102 95 96
Transportation 104 105 106 107 113 114 116 117 118 119 120 121 127 128 129 130 135 136 137 138 108 109 110 111 112 115 122 123 124
Air Quality 139 140 141 142 143 156 157 158 159 144 145 146 147 148 149 150 151 152 153 154 155 160 161 162 163 164
Solid Waste 165 166 167 168 169 170 172 173 176 171 174 175
Footprint 177 178 179 180 181 182 183 184 185 186 187
Cost Of Living 188 189 190 191 192 193 194 195 196 197 198 199 200 201
Agriculture 202 203 204 205 206 207 208 209 210 211 212 213 240 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229
Neighbourhoods 241 242 243 244 245 246
Water 259 260 261 262 263 265 266 267 268 269 270 271 272 273 274 275 276 278 279 280 281 282 283 284 285 286 287 288 289

Figure 6: The fully expanded indicator overview shows all 294 dimensions simultaneously using color encoding rather than spatial position. In this scenario, the saturated blue for most of the `Demography` indicators show a dramatic population increase. A few `Cost of Living` indicators are white, showing no change from the present day, while the rest are green, showing a decrease. (The numbers presented within each cell simply identify the indicator number and are for development purposes only.)

The small footprint of the highest-level aggregate overview, requiring only a single column of boxes, enables side-by-side comparison between multiple scenarios. Figure 8 illustrates a comparison between three scenarios. We can see in a single glance that many categories are unchanged, including `Demography`, `Solid Waste`, `Neighbourhoods`, and `Water`; the leftmost scenario has lower `Cost of Living` and `Air Quality`; and the rightmost scenario has more `Government`.



Demography 400
Economy 406
Energy 416
Government 428
Urban Growth 440
Transportation 445
Air Quality 460
Solid Waste 468
Footprint 478
Cost Of Living 484
Agriculture 491
Neighbourhoods 502
Water 507

400 403 405
406 412 413 409 411 414 415
416 419 420 422 423 424
428 431 434 438 432 435 436 437
440 443
445 451 452 454 457 459 448 450 453 455 456 458
460 466 463 465 467
468 471 474 475 477 472 476
478 481 482
484 487 488 490
491 494 501 495 497 498 499 500 505 506
502
507 512 513 514 515 517 518 519 520 524 525 526 52

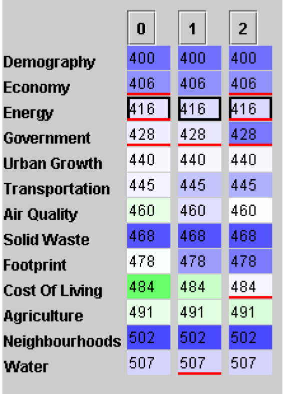
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 29 30 27 28 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
77 78 80 81 82 83 84 88 79 85 86 87
91 92 93 94 97 98 99 100 101 102 95 96
104 105 106 107 113 114 116 117 118 119 120 121 127 128 129 130 135 136 137 138 108 109 110 111 112 115 122 123 124
139 140 141 142 143 156 157 158 159 144 145 146 147 148 149 150 151 152 153 154 155 160 161 162 163 164
165 166 167 168 169 170 172 173 176 171 174 175
177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200 201
202 203 204 205 206 207 208 209 210 211 212 213 240 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229
241 242 243 244 245 246
259 260 261 262 263 265 266 267 268 269 270 271 272 273 274 275 276 278 279 280 281 282 283 284 285 286 287 288 289

Figure 7: Multilevel indicator browsing. The highest level shows aggregate values for each category, the middle level shows only the most important indicators within a category, and the fully expanded overview shows all 294 indicators simultaneously. The user can drill down by clicking on an indicator to see detailed bar charts.

3.3 Dimensionality Reduction for Context

A QUEST scenario can be considered as a point in a high-dimensional space of 294 dimensions, one for each indicator, which we will call *scenario space*. Although scenario space has far too many dimensions to inspect directly, we wanted to show how a scenario fit into the context of other scenarios by using a dimensionality reduction approach to create a two-dimensional overview. Dimensionality reduction relies on the idea that some data sets measured in a high-dimensional space have a true structure that can be closely approximated in a space of much lower dimension.

In QuestVis, we used multidimensional scaling to create a two-dimensional overview of scenario space, as shown in the upper middle pane of Figure 4. In multidimensional scaling, points are placed by minimizing the difference in distances between pairs of low-dimensional points versus their high-dimensional counterparts. This computation is time-consuming, so we pre-computed a single layout for scenario space offline using an existing program, and saved the 2D location values for each scenario point in a database. We used Hive [14], which could compute the layout for the nearly 120,000 points in 294-dimensional space in under three hours.



	0	1	2
Demography	400	400	400
Economy	406	406	406
Energy	416	416	416
Government	428	428	428
Urban Growth	440	440	440
Transportation	445	445	445
Air Quality	460	460	460
Solid Waste	468	468	468
Footprint	478	478	478
Cost Of Living	484	484	484
Agriculture	491	491	491
Neighbourhoods	502	502	502
Water	507	507	507

Figure 8: The high-level view has a small footprint, allowing side-by-side comparison between columns of multiple scenarios. This view is a true aggregate of all 294 indicators, as opposed to the radar graph in Figure 2 which only shows 9 of the indicators.

Each scenario point is always shown in the same spatial location to create a stable spatial substrate, and the user can interactively change select which one of the 294 indicator values should be used for color coding the points. Figures 9 and 10 show close-up views of the scenario space overview pane, where the `Time in Car` indicator was chosen to color the former and `Industrial Energy Use` was selected for the latter. Figure 10 also shows the visual history mechanism, a trail marking the twelve most recently selected scenarios in the interactive session.

The layout of points in the overview pane is dense enough that many points may be overlapped in the same screen pixel. The overview window has a slider that filters the number of points shown based on the value of the currently selected indicator that is used to color the points. Figure 11 shows an overview colored by `Domestic Water Use`, filtered to show only scenarios with the highest values for this indicator.

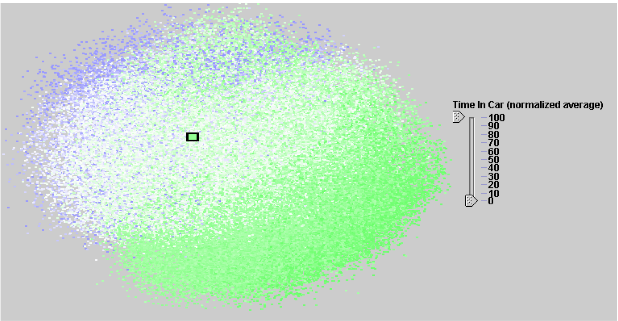


Figure 9: Dimensionality reduction to show where a particular scenario, marked by the rectangle, fits within the context of scenario space. Here, the points are colored by the value of the `Time in Car` indicator.

4 Reflections on the First Prototype

We obtained feedback on the first interactive prototype from the expert modellers and workshop facilitators at SDRI and Envision.

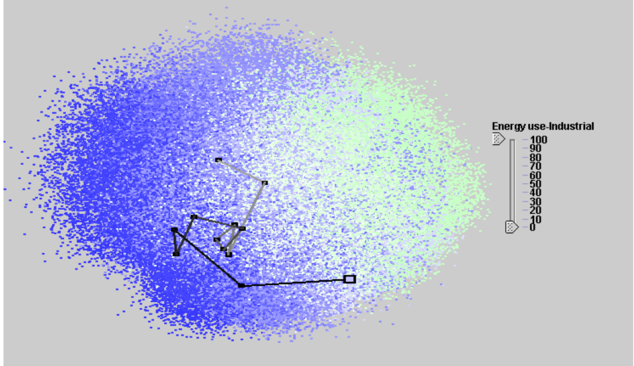


Figure 10: Scenario space overview colored by the `Industrial Energy Use` indicator, with trail showing the history of previously selected scenarios.

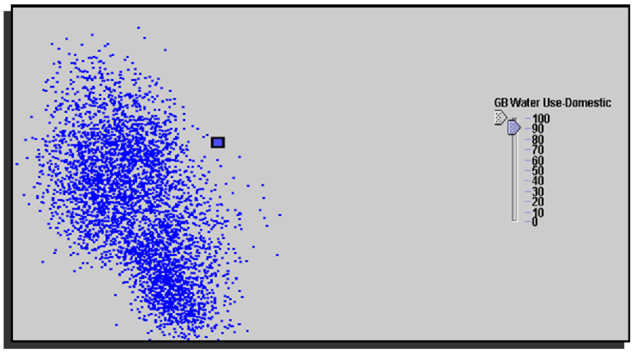


Figure 11: Scenario space overview colored by the `Water Use` indicator, and filtered to only show scenarios with the greatest increase for this indicator.

4.1 Immediate Response

Providing an immediate response to input changes at any time allowed exploratory ”What if?” queries and exploration of many different future scenarios. The architectural change of using a precomputed database was considered highly successful. This idea was immediately incorporated by Envision into their main product line. Their current MetroQuest system is now built on a precomputed database.

4.2 Dimensionality Reduction

The use of dimensionality reduction was deemed to cause far more confusion than insight. The scenario space overview was an incomprehensible cloud of dots. The failure came from two quite distinct reasons: a mismatch with the characteristics of the target audience and task, and the mathematical properties of this particular data set.

First, the idea of a low dimensional embedding of a high dimensional space required far too much time to communicate to a nontechnical user during a brief session. The target audience does not necessarily have a scientific or mathematical background and were only intending to use our software for a short time, from a few minutes to a few hours. The amount of time available for explaining this aspect of the the software was thus only a few minutes at best. Following an explanation on the meaning of the overview from a facilitator required multiple conceptual leaps on the part of the audience: defining an indicator of future life as an abstract dimension, considering a high-dimensional space beyond the familiar three dimensions of direct perception, and embedding that high-dimensional space to a two-dimensional plane. Moreover, one of the project goals was to allow the software to be used without a facilitator. The experts considered it highly unlikely that the audience would understand the meaning of the scenario space overview through interactive exploration with the software alone, without explanation from a facilitator.

The past successes of dimensionality reduction in visualization have been in very different circumstances: target audiences of scientists, who would be doing analysis for extended periods of time. This project began before the publication of an intriguing definition of *casual* information visualization in opposition to expert use for deep analysis tasks [13]. Using that vocabulary, our usage scenario is a better match with casual infovis than traditional infovis.

Second, the amount of information lost in the reduction from 294 to two dimensions was so great that very little information could be reliably inferred from the scenario space overview, even for an observer who completely understands the mathematical ideas behind the visual representation. The intent of the display was that similar scenarios would be clustered together, with dissimilar scenarios far apart. Dimensionality reduction is typically useful when points fall into spatially distinct clusters in the low-dimensional embedding. However, in the scenario space overview, the points were fairly evenly distributed in an oval region. A scenario's location in 2D space was very difficult to predict, and the trail of explored scenarios was close to a random walk. We concluded from these results that the true dimensionality of this data set was considerably higher than two. Later analysis using MATLAB showed that five dimensions were required to characterize most of the variance in the data set, and 18 dimensions were enough to characterize nearly all of it. Several techniques, including using a matrix of scatterplots rather than a single scatterplot, have been proposed for analyzing dataset of higher true dimensionality than two. However, all these would require even more explanation, and so would also be inappropriate for a casual use scenario. We thus decided to abandon a dimensionality reduction approach for this problem.

4.3 Multilevel Indicator Browsing

The multilevel indicator browsing had mixed reviews. While the ability to compare indicators side by side between multiple future scenarios seemed promising, the representation was so compact that comprehensibility suffered. The labels for output indicators were tooltips available on mouseover for an individual cell. Examining indicators one a time slowed down detailed exploration and provided little inspiration for further examination. Although the relationship between the most compact high-level view and the fully expanded view was clear, the mid-level view was confusing because it was unclear what the aggregate values represented.

4.4 Aggregation and Normalization

Averaging multiple low-level cells into a single aggregate higher-level cell often did not reflect a valid summarization of a category. We had carried out a first level of normalization separately for each indicator, in hopes of allowing sensible aggregation despite differences in the original scales or units of the indicators. The normalized scale ranged from the maximum possible decrease between the present and future to the maximum possible increase.

However, this method of aggregation treats all indicators neutrally. The aggregate value shows whether the contained items increased or decreased on average. After using the interactive prototype, our collaborators noted that many indicators had conflicting values in the eyes of any particular person, with some positive and some negative. Aggregation that did not reflect those personal values was not very useful.

For example, the `Cost of Living` category shown in Figure 12 contains the indicators `Average Wages` and `Tax`. When average wages increased and tax decreased, the aggregate cost of living indicator showed a neutral value because the signs of the indicators are opposite. Intuitively, most users would consider both of these changes an improvement, and expect that the scenario should show improvement in the aggregate value.

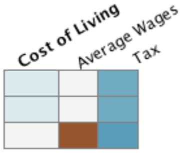


Figure 12: Averaging across indicators that are ascribed opposite values by users led to unintuitive aggregate displays.

4.5 Aesthetics

The look and feel of the first prototype was extremely primitive compared to the polished interface of QUEST. Although research prototypes are typically held to a lower standard of graphic design than commercial software, interfaces designed for the general public do need a certain level of friendliness and visual appeal. We deemed it important to pay closer attention to aesthetics in the next attempt.

5 The Second Prototype

The first prototype was designed to dramatically increase the density of information simultaneously visible to the user, but the abstraction was so extreme that the tool was deemed difficult to engage with by casual users. For the second prototype, we wanted to keep the interactive and exploratory nature of the first prototype, while addressing its limitations. We completely eliminated the scenario space overview. For multilevel browsing, we kept the high-level view and eliminated the mid-level view. We changed the low-level view to show one category at a time, with visible labels, and be visible simultaneously with the high-level view to support examining details of a scenario while still comparing the summary information with other scenarios. Finally, we improved the overall look and feel to provide a friendlier interface suitable for a casual user.

Figures 13, 14, and 15 show the second prototype when exploring one, two, and five scenarios. At the top, the axes allow the user to change input choices, and trails showing the selections for all active scenarios are simultaneously visible. The middle layer shows high-level summarized scenario views side by side for easy comparison. The bottom layer shows drilled-down views of a chosen indicator category in detail, again for every active scenario.

5.1 Scenario Generation

The input region at the top of the window acts both as a control of and a display for multiple active scenarios. A query is formed by choosing one option for each input choice. The radio button design of the previous prototype could show only one active scenario, with an external mechanism to show history only as trails in the abstract scenario space. In order to establish a stronger link between input choices and future scenarios, this prototype was designed to display the choices for multiple scenarios at once, providing a rich history. The design was motivated by parallel coordinates, which are are

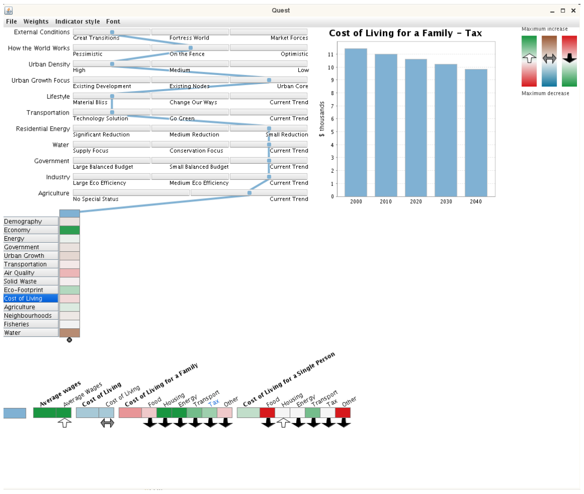


Figure 13: Browsing one scenario in the second prototype.

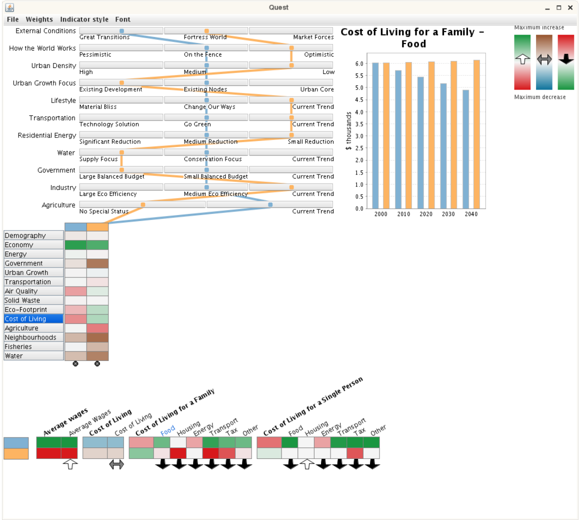


Figure 14: Comparing between two scenarios in the second prototype.

normally used to represent points in continuous multidimensional space [7, 18]. Unlike Cartesian coordinates, where each new axis is placed orthogonally to all other axes, each axis is placed parallel to all other axes. 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 interface design was inspired by the work of Tory *et al.* [16]. Each axis consists of active buttons placed side by side horizontally. All of the choices for an individual scenario are linked by a colored line that zigzags through the input axes, and a colored dot marks each of the choices. When creating a new scenario the dots appear after clicking on a button, or the user can modify an existing scenario by dragging the dot to a new location. The scenario line continues past the bottom axis to a block marked in the same color at the top of the high-level output indicator summary for that scenario, in the middle of the window. The combination of the joining line and the common color creates a strong perceptual linkage between inputs and outputs.

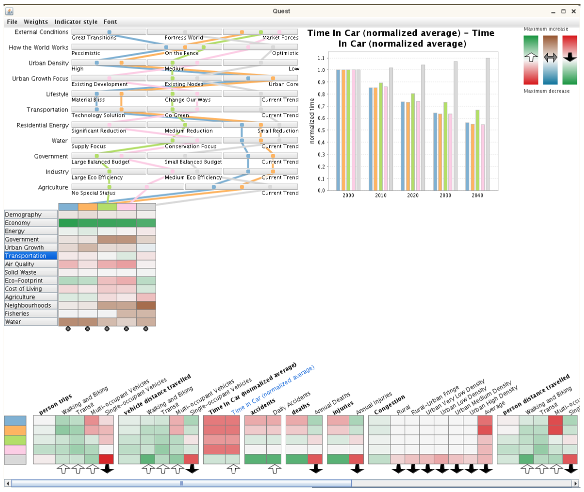


Figure 15: Comparing between five scenarios in the second prototype.

Our use of color is informed by the human perceptual limitations of the small number of distinguishable colors that can be used to represent nominal values in small spatially separate regions. The prototype has a carefully chosen pallette of eleven unique colors. If more than eleven scenarios are active at once, the colors are reused in the same order. Confusion is unlikely because the lines connecting the the dots creates a strong perceptual grouping, and the dots are ordered horizontally on the buttons so there will always be ten other dots separating a repeated color.

5.2 Two-Level Indicator Browsing

In the middle of the window, each scenario is summarized in a column showing a aggregate view of the the fourteen high-level categories. Each row has a label on the far left, and a colored box representing the aggregate values for the category in each scenario. Summary columns can be rearranged with drag and drop, with the scenario lines from above automatically following to the new positions, so that any two scenarios can be placed side by side for easy pairwise comparison.

Clicking on a labelled category button highlights it in blue, and changes the expanded view at the bottom of the window to show that category. In the expanded view, all of the indicators for the chosen category are shown as colored boxes, with a row for each active scenario. These rows are visually linked with the scenario lines above using a header box of the same color on the far left. The vertical order of the scenario rows in the detail view is the same as the horizontal order of the draggable scenario columns in the summary view, to maintain context.

In the expanded view, a column shows the same indicator across multiple scenarios, and a text label for the indicator appears at the top. In order to have densely packed yet readable text, the labels are drawn on a diagonal rotated 30° from the horizontal. With this layout, it is easy to scan the labels for measures of interest and then make comparisons down the columns. We chose to arrange the scenarios in rows for the detail view to take advantage of the aspect ratio of most monitors, assuming higher resolution horizontally than vertically.

Clicking on the colored box for any indicator in the expanded view shows its full details in a bar chart in the top right corner, with color-coded bars for each active scenario.

5.3 Aggregation

As discussed above, for some indicators an increase is an improvement, while for others a decrease is an improvement. A technically straightforward solution would be simply record a choice for the polarity of each indicator in the master database. For some indicators, such as automobile deaths per year, it is trivial to assign the meaning that increased traffic deaths are bad. However, for many indicators, assigning a good or bad value is controversial. Is increased nuclear power good or bad? What about high-density housing, social services spending, or bus trips to work? The goal of the QUEST project is to promote discussions about values amongst the users of the tool, rather than advocating the specific answers dictated by the tool creators. Furthermore, there are several indicators, such as population age demographics, that can be considered neutral.

We therefore allow the users to interactively make and change personal value judgments of positive, negative, and neutral for each indicator in the tool interface. In the bottom detail pane, each indicator has a simple arrow below it to indicate whether the indicator is currently considered good, bad, or neutral. A simple mouse click will change the mapping, cycling through the three choices of the white upward good arrow, the black downward bad arrow, or the grey double-sided horizontal neutral arrow. As soon as an indicator mapping is changed, there is an immediate update of the color scale used and any upstream aggregations involving that value. Figure 16 shows a close-up example of indicators and their arrows.

The second prototype uses one of three color scales, depending on the personal value mapping for the indicator. We use a culturally familiar red-green divergent scale for the positive indicator, and the same color range inverted for the negative indicator, so that good indicators are green and bad are red. We use a visually distinguishable blue-brown divergent scale for neutral indicators. A legend is visible at all times in the upper right corner of the window to help users remember the meaning of the color coding. When aggregating a mix of both neutral and good/bad indicators, we use the majority to choose which kind of arrow to show.

The entire set of current values can be saved for later reloading. Thus, users do not have to explicitly make choices for values before they start using the program. Using a pre-loaded set of values does not prevent the user from customizing individual values while exploring future scenarios. A starting set of values could be provided by a facilitator in a workshop setting, or be automatically generated from the results of a separate values quiz taken by a user before using the prototype.

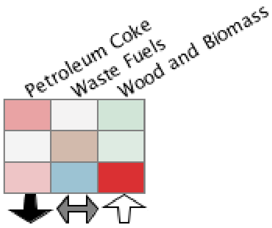


Figure 16: Close-up view of indicators with clickable arrows beneath showing the choice of deteriorating, neutral, and improving values from left to right.

6 Reflections on the Second Prototype

We again obtained feedback from modellers and facilitators at SDRI and Envision on the second prototype.

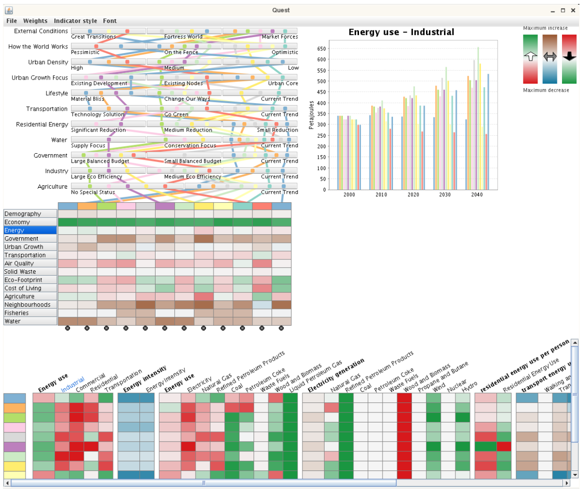


Figure 17: At 12 scenarios, the second prototype nears the limit of manageability.

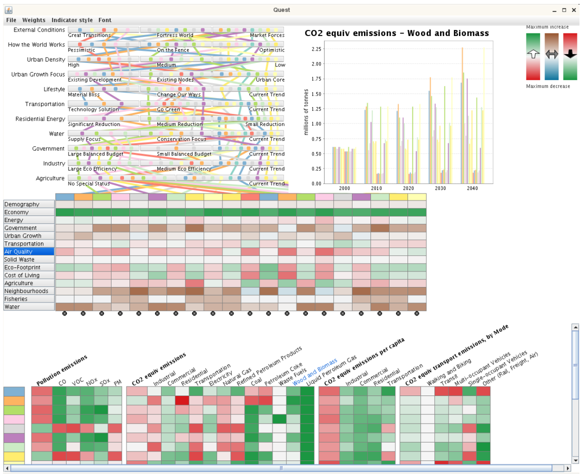

Quest
File Weights Indicator style Font
CO2 equiv emissions - Wood and Biomass
millions of tonnes
2000
2010
2020
2030
2040
Maximum increase
Maximum decrease
Pollution emissions
CO2 equiv emissions
CO2 equiv emissions per capita
CO2 equiv transport emissions, by Mode

Figure 18: With 20 scenarios, the second prototype is clearly unwieldy.

6.1 Successes

QuestVis2 was considered to be much more understandable than the first prototype. The combination of the scenario generation display and the two-level overviews supported both browsing through the indicators for a single scenario and comparing between multiple scenarios. Allowing users to express their own values in order to see meaningful aggregation in the overviews was considered to be a notable success. The look and feel were much cleaner in the second prototype.

6.2 Scalability

The second prototype works well for browsing through a single scenario and comparing a limited number of scenarios. When too many scenarios are active at once, the scenario lines are hard to follow and scrolling is required to browse the detailed category outputs. Figure 17 shows that viewing 12 scenarios simultaneously nears the limit of manageability, and 18 shows that viewing 20 scenarios once is clearly unwieldy.

There is a lot of unused screen space when browsing a single scenario or a small number of scenarios. If we use a target of 10 or fewer scenarios, it would be straightforward to improve the information density by simply rearranging the major components within the window and redistributing the space allocated to them: the scenario space generator, the summary columns, the category view, the detail view, and the color legend.

6.3 Problem Mismatch

Although the design goals for the second prototype as we understood them were met, the QUEST designers were still uneasy about deploying the tool. After further discussion, we came to understand that there was a fundamental mismatch between our view of the problem and theirs. We had approached the problem with the intention of creating software that would help its users develop a mental model of the complex relationships between the input policy choices and the output indicators. However, the intuition of the facilitators after using the second prototype was that it exposed far *too much* of the complexity of the model to users. Their practice in leading groups through the QUEST software was to deliberately limit the complexity by focusing on just a few key choices and indicators. We had thought that this limitation was part of the problem that our software should help solve, and designed the prototype to help users explore the rich information space of the high-dimensional data. It was only after the post-QuestVis2 discussions that we realized that the designers of QUEST did *not* want users to focus too much on the internal details of the model. The high-level goal of the QUEST project is to promote environmental sustainability and behavior change via community engagement in environmental policy decisions. Teaching the general public about the information space underlying their model is thus a sidetrack and a diversion from this main goal.

The idea that exploring an information space is a central user goal is a typical bias of an information visualization point of view. In this case, it was not the right match with the problem at hand. After we understood this, we halted the project and did not deploy or further refine the QuestVis prototype.

6.4 Visualization Design Process

We recently introduced a model of the visualization design process with four nested stages: domain problem characterization, data and operation abstraction design, visual encoding and interaction design, and algorithm design [11]. Some aspects of that model were inspired by reflection on the evolution of this project.

Using the language of that model, we see in retrospect that this project ran aground because of a misunderstanding at the first stage of domain problem characterization. The cascading implications of that misunderstanding are that we were unable to choose the right abstraction at the next stage, and thus that refinement at the third stage of visual encoding could never fix these fundamental upstream flaws. For the first prototype, we chose dimensionality reduction as a visual encoding technique. For the second prototype, we backed away from dimensionality reduction but still framed the underlying abstraction as exploring a high-dimensional information space. The visual encoding we designed did successfully support that abstraction, but that abstraction does not address a valid user need.

7 Related Work

The first QuestVis prototype is discussed in more detail by Williams in his thesis [20]. The two prototypes make use of several information visualization techniques previously introduced in the literature. The utility of linked views has long been established [2], and previous systems in many application domains have been built around this approach [1, 6, 9, 10, 15].

The multilevel indicator browser was inspired in part by previous interfaces that use color as a compact way to represent data, such as the heatmaps currently popular in the bioinformatics literature [19]. The perceptual and cognitive tradeoffs between a compact representation using color and a more traditional representation using spatial positions have since been studied, both in the context of this particular application [5] and in more general settings [8].

We did not break new algorithmic ground in dimensionality reduction techniques, since we use an existing algorithm for layout [14]. However, our experience may shed some light on the question of when dimensionality reduction methods are appropriate, a topic that is still largely unexplored in the previous literature.

8 Conclusion

We have discussed the lessons learned in building a visualization tool intended for exploration of a dataset generated by an environmental sustainability model. The use of linked views and aggregation to create a compact overview were appropriate techniques for visual encoding and interaction given the design goals of the project. However, those goals were based on a mischaracterization of the problem at hand: providing the target users with the analysis tools for a high-dimensional dataset was in fact counterproductive in the larger context of the sustainability project. This lesson was a motivating factor in the creation of a new nested model for the design and validation of visualization system.

9 Acknowledgements

We are very grateful to the many people involved with QUEST for the time and energy they devoted to this project, especially John Robinson of SDRI, Mike Walsh and Dave Biggs of Envision, and Jeff Carmichael of Envision and SDRI. This work was supported by grants from the NSERC-funded GEOIDE NCE: “CIRCUITS: Collaborative Interdisciplinary Research on Communities Using Information Technology for Sustainability”, and “GeoCognito: Connecting People with Ideas and Ideas with Place”.

References

[1] Aaron Barsky, Tamara Munzner, Jennifer Gardy, and Robert Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis ’08)*, 14(6):1253–1260, 2008.

[2] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[3] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 91)*, pages 181–186, 1991.

[4] Jeff Carmichael, James Tansey, and John Robinson. Georgia Basin Quest: an integrated assessment modelling tool. *Global Environmental Change*, 14(2):171–183, 2004.

[5] Cristina Conati and Heather Maclaren. Exploring the role of individual differences in information visualization. In *Proc. Advanced Visual Interfaces (AVI)*, pages 199–206, 2008.

[6] Paul Craig, Jessie Kennedy, and Andrew Cumming. Animated interval scatter-plot views for the exploratory analysis of large-scale microarray time-course data. *Information Visualization*, 4(3):149–163, 2005.

[7] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. IEEE Visualization (Vis)*, 1990.

[8] Heidi Lam, Robert Kincaid, and Tamara Munzner. Overview use in multiple visual information resolution interfaces. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 07)*, 13(6):1278–1285, 2007.

[9] Heidi Lam, Daniel Russell, Diane Tang, and Tamara Munzner. Session viewer: Visual exploratory analysis of web session logs. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 147–154, 2007.

[10] Alan MacEachren, Xiping Dai, Frank Hardisty, Diansheng Guo, and Gene Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 31–38, 2003.

[11] Tamara Munzner. A nested process model for visualization design and validation. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis '09)*, 15(6):921–928, 2009.

[12] M. Plumlee and C. Ware. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *Proc. ACM Trans. on Computer-Human Interaction (ToCHI)*, 13(2):179–209, 2006.

[13] Zachary Pousman, John T. Stasko, and Michael Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis 07)*, 13(6):1145–1152, 2007.

[14] Greg Ross and Matthew Chalmers. A visual workspace for hybrid multidimensional scaling algorithms. In *Proc. IEEE Symposium on Information Visualization (InfoVis 03)*, pages 91–96, 2003.

[15] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005.

[16] Melanie Tory, Simeon Potts, and Torsten Möller. A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11:71–80, Jan./Feb. 2005.

[17] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

[18] Edward J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, September 1990.

[19] Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.

[20] Matt Williams. QuestVis and MDSteer: The visualization of high-dimensional environmental sustainability data. Master's thesis, University of British Columbia, Department of Computer Science, July 2004.