# Inventing Information Representations Through Task Analysis

Nicholas P. Wilde and Clayton Lewis

CU-CS-549-91 October 1991

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

# Inventing Information Representations Through Task Analysis

Nicholas P. Wilde and Clayton Lewis

CU-CS-549-91 October 1991

Department of Computer Science
Campus Box 430
University of Colorado,
Boulder, Colorado, 80309

This research is supported, in part, by US West Advanced Technologies.

# INVENTING INFORMATION REPRESENTATIONS THROUGH TASK ANALYSIS

*Nicholas P. Wilde and Clayton Lewis*

Department of Computer Science
Campus Box 430
University of Colorado, Boulder CO 80309-0430

wilde@cs.colorado.edu
clayton@cs.colorado.edu

**ABSTRACT**
Progress in HCI is often progress in the representation of information. We present a method for designing new representations based on an old trio of friends: Task analysis, early prototyping, and user testing. To illustrate the method, we present a case study of the design of a representation for a business modelling task. This task combines the need for both quantitative and qualitative types of information. Besides illustrating the technique developed, the case study presents some novel representation techniques of interest in their own right.

**KEYWORDS:** Information representation; Design Process; Task Analysis; Case Study

**INTRODUCTION**
Major progress in HCI can often be equated with progress in representation, in a broad sense. Breakthroughs like the word processor and the spreadsheet centered on new ways of mapping objects and tasks of interest onto displays and operations on them. Such designs, successful though they have been, hardly begin to tap the possibilities of computerized representation, as recent work by Card, Mackinlay, and Robertson [1,7,10], exploring the use of three dimensional depictions, animation, and color, has shown. If we look at the technical resources now becoming available we can expect a progression of exciting new techniques. But where are we going to get the ideas we need to apply these techniques to useful work? The spreadsheet did not result simply from exploiting the potential of the microcomputer display as a medium; rather, it required insight into the task of financial analysis and how it could be supported [5]. We may feel that this insight, and the accompanying invention, are part of a creative process on which HCI is reliant as a discipline, but to which it can contribute little. We as HCI professionals may be good at tidying up designs, or in making helpful suggestions about development methodology, but we didn't invent the word processor or the spreadsheet.

In this paper we argue for a much more optimistic view: We can take systematic action to promote the invention of new, useful representations. We first summarize the conceptual foundations of the design of representations as developed by Mackinlay, Casner, and others. We then point to the centrality of task analysis in the application of these ideas, following Casner. We describe a practical method for integrating task analysis into the design of representations. We describe experience in applying this method to a problem in modelling complex business operations. The method led to the invention of a number of novel representation techniques. While the effectiveness of these techniques remains to be established, they are of sufficient interest to indicate the value of the method we present.

**THEORY OF REPRESENTATIONS**
In a seminal paper Mackinlay [6] developed a conceptual framework for the problem of representation capable of describing formally how graphs depict relational information, and of permitting the automatic construction of new combinations of graphical conventions to form graphical representations for particular applications. While Mackinlay's work focussed on how the nature of the information to be represented determines the suitability of the representation, Casner [2] (see also Roth and Mattis [12], and Robertson [11]) focussed on how the task users wish to perform determines suitability, noting that the same information may need to be represented in different ways to support different tasks.

The work of Mackinlay, Casner, and others can be seen as an extension of earlier work in the theory of measurement [4] (see also Wehrend and Lewis [13]). In the framework developed there, a representation consists of a mapping between an application domain of interest and a representation domain. Relations and operations of interest in the application domain are mapped to relations and operations in the representation domain. A representation is good only if the resulting operations in the representation domain are easy to perform. For example, if the application domain consists of objects with various temperatures, we might construct a representation in which line segments of various lengths represent the objects, with the relation "hotter than" being mapped to the relation "longer than".

The operation of determining the hotter of two objects is mapped onto the operation of determining which of two lines is longer. If the lines are arranged so that they are parallel and have one end aligned, this operation will be easy and the representation will be suitable. But if the lines are arranged with differing orientation and alignment the representation will not be suitable, because the operation we are interested in is difficult. To carry the example further, if we wanted to be able to find an object with median temperature we might want the line segments to be sorted by length.

In Mackinlay's terminology, this example brings out the distinction between two attributes of a representation. A representation satisfies an *expressiveness* criterion if operations in the representation domain accurately reflect facts about the application domain. A representation satisfies an *effectiveness* criterion if the relevant operations in the application domain are easy. The example also illustrates Casner's point: devising a good representation depends not only on the nature of the information to be represented but also on what operations on the data users want to perform. This brings an analysis of user tasks into the center of the problem of designing representations.

Casner [2] presents a *logical task description language* in which user tasks can be formally described as sequences of logical operations that make no reference to a graphical representation. His BOZ system can then automatically transform this description into a procedure for performing the task as aided by a graphical representation. The graphical representation is developed by finding opportunities to replace difficult steps in the logical task description by easy perceptual operations on graphical entities.

Casner's method, like that of Mackinlay, aims to formalize the workings of representations so that representations can be built up automatically as combinations of specified primitives. The general logic of the approach, however, can be applied to the broader problem of devising representations that are not constrained to be built from known primitives and need not be accessible to an automated method. In our method, like Casner's, we describe a sequence of mental operations adequate to perform a selected task. The difficult steps in the sequence are identified, and representations are sought that can support these steps. But we do not require that the process of proposing representations for these steps be automated or that the knowledge involved be formalized.

## OUTLINE OF THE METHOD

The major steps in the method are summarized in Table 1. There are a number of differences from Casner's method intended to make the method more appropriate to the work of human designers, and to eliminate the dependence on already-known and formalized representation techniques. Rather than analyzing a task as it would be performed with no graphical representation, as Casner does, we choose a plausible representation and analyze the task as it would be performed using this. This requires less hypothetical reasoning. We accept an informal description of the operations needed with this representation, and an informal determination of difficulty. We bolster this informal analysis with data obtained by asking users to perform the task using the provisional representation, which we could not do if we followed Casner in starting with a representationless version of the task. We rely on brainstorming to propose changes to the provisional representation, once the design problem has been narrowed by the foregoing analysis. Any ideas are admissible. We reanalyze the task as supported by possible modified representations, and again collect data as a check on the analysis.

Table 1. Major steps in task-analysis based design of representations

| |
|---|
| 1. Choose tasks. |
| 2. Outline mental and perceptual operations needed to perform tasks. |
| 3. Collect user data as check on analysis. |
| 4. Identify specific difficulties in task. |
| 5. Invent ways to modify provisional representation. |
| 6. Reanalyze tasks as performed with new representations. |
| 7. Collect user data as check on analysis. |

## A CASE STUDY

As part of a collaborative research program with US West Advanced Technologies we were asked to contribute to the design of a decision support environment for managing overseas business ventures. This environment should include a model of business operations capable of representing not only standard financial data, but also projections, with associated uncertainty, and qualitative factors that might affect a business, such as possible changes in staff availability. Earlier work on this kind of model used two kinds of representations. The first in use was a graph model, in which quantitative and qualitative factors are shown as nodes influencing each other over links representing causal connections among them (see Figure 1). The other representation already in use was an extended spreadsheet, implemented using a combination of the Excel spreadsheet [8] and the @Risk extension package [9] (see Figure 2). In this package, factors are represented by spreadsheet cells and their interactions are encoded as associated formulae. The @Risk package permits quantities to be specified as variables with specified statistical distributions, and uses Monte Carlo simulation to determine the distribution of values for quantities derived from these.

Though these two representations permitted useful work to be done, neither was completely satisfactory. Broadly, it was felt that the graph model was good for representing interactions among qualitative factors but would need to be extended to handle quantitative financial data adequately, while the extended spreadsheet had the opposite attributes. We were asked to explore the possibility of extending

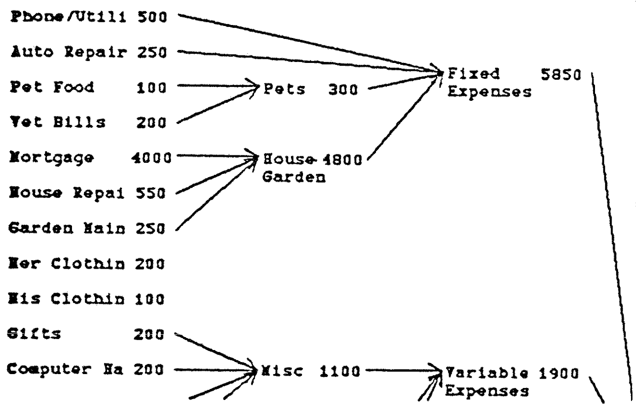


Figure 1. A portion of graph model, representing a household budget.

| | |
|---|---|
| Phone and Utilities | $500 00 |
| Auto Repair | $250.00 |
| *Pet Food* | $100.00 |
| *Vet Bills* | $200 00 |
| Pets | $300 00 |
| *Mortgage* | $4,000.00 |
| *House Repair* | $550.00 |
| *Garden Maintainance* | $250 00 |
| House and Garden | $4,800.00 |
| Fixed Expenses | *$5,850.00* |
| Her Clothing | $200 00 |
| His Clothing | $100 00 |
| *Gifts* | $200.00 |
| *Computer Hardware* | $200 00 |
| *Books* | $200.00 |
| *Garden Additions* | $200.00 |
| Misc | $1,100 00 |
| Food | $480.00 |
| *Entertainment* | $200.00 |
| *Dining Out* | $200.00 |
| *Entertain At Home* | $100.00 |
| Leisure Activities | $500.00 |
| Variable Expenses | *$1,900.00* |
| *Total Expenses* | $7,750.00 |

Figure 2. The same model, represented as an extended spreadsheet.

either representation, or creating a new representation that would have the virtues of both.

### Choosing Tasks

To determine how these representations might be improved we needed to choose tasks that would permit us to study how the representations did or did not support user needs. Because no operational system now exists we first discussed user requirements with the designers of the proposed system, who were themselves potential users and who were in touch with other potential users. We then listed what we saw as the strengths and weaknesses of the representations already in use, with respect to these user requirements. From this, we defined a set of tasks that covered the user requirements well while at the same time illuminated the apparent contrast between the existing representations. These were generic tasks, from which a set of model specific tasks could be derived for a particular model. Table 2 is a sampling of the generic covering tasks and model specific tasks derived from them.

In another situation task selection might begin and end simply with user requirements, without considering what more abstract issues might separate one possible representation from another. In our case, considerable work by others had already gone into the existing representations. We felt it was important that our analysis should help to understand the particular strengths and weaknesses of those representations as well as proposing a modified one.

### Choosing a Provisional Representation

In our situation a provisional representation was given. We actually carried along both existing representations as provisional representations in our analysis. In other situations, representation designers might have to work a little harder to find provisional representations, although in our experience there are usually adequate sources of provisional representations available. Good sources include looking at the way similar tasks have been done in the past, at potential users existing models and diagrams of their tasks, and at existing general purpose representations such as spreadsheets or dataflow diagrams.

### Outlining Mental and Perceptual Operations needed to perform Tasks.

We made a step by step analysis of how each of our tasks would have be accomplished in the existing representations. As it happened, our tasks appeared to be recombinations of a few standard subtasks, so that we did not need to analyze each task anew. These recurring subtasks are listed in Table 3.

Table 2. A sampling of generic covering tasks and model specific instantiations of them

| | |
|---|---|
| Anticipate the consequences of an event . | Spring is unusually wet & cold. How is the budget effected ? |
| Explain an event internal to the model. | Your "fixed" expenses mysteriously doubled last quarter, proving that they weren't fixed after all. Give some possible explanations for this. |
| Plan for an event internal to the model. | House repairs are needed for the third quarter. How can you compensate for this and keep the budget balanced ? |
| Alter a model - add a node/cell to the model. | Add an event labelled "Have a child" |

Table 3. Recurring subtasks that arose from task analysis

| |
|---|
| **Find cells/nodes of interest** – find cells/nodes in the model that seem to applicability to the problem at hand. |
| **Reason forward from causes to effects** – given a cell/node of interest, reason forward about possible consequences of changing the value of that cell/node |
| **Reason backward from effects to possible causes** – given a change in the model at a particular node/cell, reason backwards about possible precedents that could have causes the observed effect. |
| **Reason sideways** – use a combination of forwards and backwards reasoning to identify possible compensatory strategies for changes observed to take place in the model. |

For each of these subtasks we outlined how it would be performed using the two provisional representations. For example, tracing the effects of changing a factor required following arcs in the graph representation and obtaining a list of dependent cells in the spreadsheet representation.

**Collecting User Data as Check on Analysis.**
We did not think it practical to educate test participants about the intended business modelling application. Instead we chose three of our generic tasks that provided good coverage of the key subtasks identified in our analysis and transposed them into a household budget application, as shown in Table 2. We then built a small model of a household budget in each of the two representations; portions of these models are shown in Figures 1 and 2. We asked six users to perform the three tasks under thinking-aloud instructions.Three users performed the tasks with the graph representation and then with the spreadsheet representation, while the other three worked first with the spreadsheet.

We felt that user behavior was reasonably consistent with our analysis, with one important exception. Factors in a causal model can have either a positive or negative influence on other factors. Answering a specific question about how events in the real world are reflected in the model requires that users be able to tell how a factor is encoded. In particular, users were asked to work out the effects of bad weather on our household budget. They could see that the factor "weather" had a "positive" influence on "utilities." But what exactly does this mean? One possible interpretation is that the units measure "amount of bad weather" – another is that they measure "goodness of weather." No explicit information about the encoding was available in either representation. Our analysis of the task simply left out this determination, and hence missed a problem with both representations.

**Identifying specific difficulties in the tasks.**
Our task analysis, together with observations of test users, suggested the following difficulties with the provisional representations.

*Spreadsheets don't show causal connections*
In the spreadsheet representation, causal connections among factors are hidden in the formulae. This meant users could not pick up information about these connections unless they looked for it explicitly. Users using the graph representation were observed picking up useful information almost by accident. While following one node-arc path a user would notice another arc nearby, or connected to the same node, that provided further insight. Furthermore, although information was available in the spreadsheet about causal connections (one could specifically select cells that were precedents or dependents formula-wise to a selected cell), the information was difficult to get at, and could only be viewed for one cell at a time.

*Graph representations take up too much space*
The graph representation suffers from not being as compact as the spreadsheet representation. The household budget model extended across four computer screens in graph form, but fit onto one screen in spreadsheet form. Spreadsheet users were seen to grasp more readily the overall structure of the model, which was important in finding the part to work on for a given task. Some users made errors attributable to not realizing there was additional information off the screen.

*Neither provides enough information at nodes/cells*
Neither representation provided adequate descriptive information about the factors. As noted earlier, neither included information about how factors were encoded, but we judged that other kinds of descriptive information, including a longer statement of the meaning of a factor than can be included in a cell or node name, would also be useful. Users also uniformly wanted more explanations as to why one factor affected another.

The tendency of just about all users to want "reasons why" *x* affected *y* was reflected in a propensity to make up stories to themselves. Such behavior persisted even in the face of contradictory evidence presented in the representation. This again called out the very strong need for a greater quantity of information at each node in the graph – what the node actually stood for, how the value was affected by other nodes with edges coming into it, how it affects other nodes in the graph. The need for such explanatory information was also evident in the spreadsheet, although perhaps because the linkages weren't as explicit in the spreadsheet, users spent less time worrying about how things were connected to each other, and the need for explanations wasn't as strong.

**Invent ways to modify provisional representations.**
These problems provided an agenda for invention. Could we find ways to represent causal connections that would be as available as arcs, and as well integrated with the overall display of the model, but would take less space? Could we find ways to present more descriptive information about the factors without taking up more screen space?

We devised three new ways of representing causal connections compactly. The *coloring* scheme (Figure 3) adds color to a spreadsheet display to mark cells with

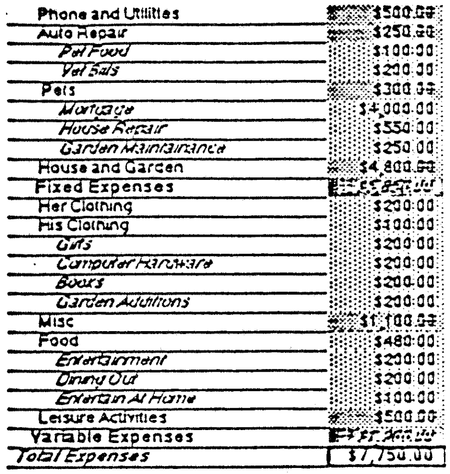


Figure 3. The *coloring* scheme designed to show causal links on a spreadsheet. Varying dot densities represent the varying distance from the "Total Expenses" cell.

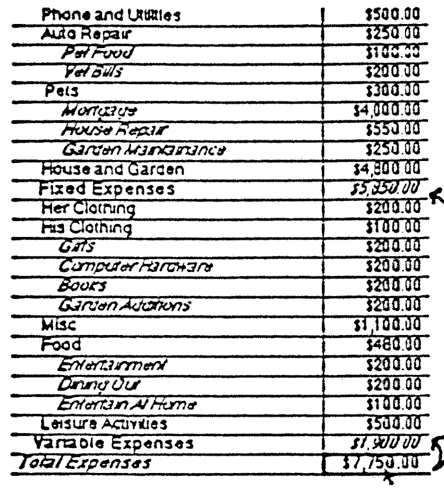


Figure 4. The *Arcs on Demand* scheme designed to show causal links on a spreadsheet. Arcs are drawn to precedent or dependent cells from the currently selected cell.

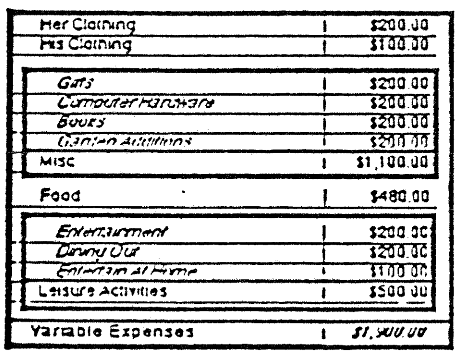


Figure 5. The *special forms* scheme designed to show causal links more compactly. The items within each box represent a sum – the item in the last row of the box represents the total.

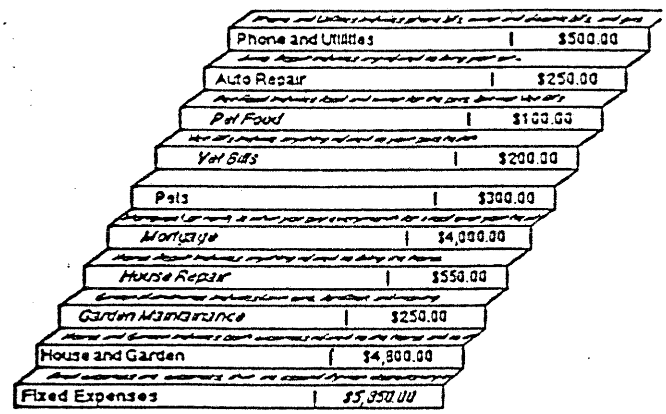


Figure 6. The *Stairways* Scheme designed to increase the information available at each node or cell. Each cell represents the front "face" of a staircase. Users can access information on the top faces by "tilting" the staircase toward themselves.

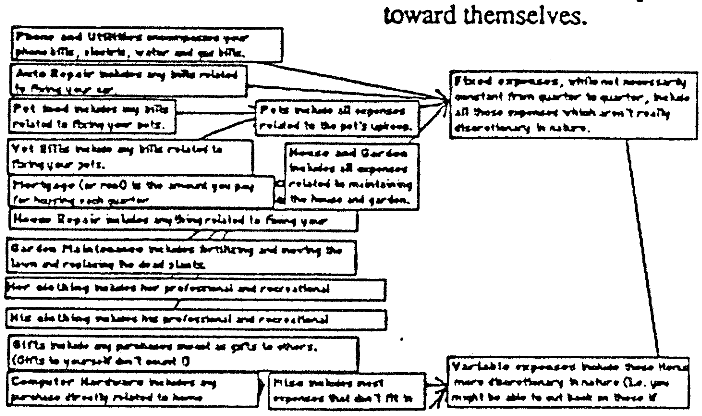

Figure 7. The *Tiling* scheme designed to increase the information available at each node on a graph model. Space usually taken up by the arcs of a graph representation is temporarily taken over by the expanded view of the nodes.

causal links to or from a selected cell. Shades of decreasing saturation would be used to distinguish cells directly linked to the selected cell from those two links away, or three, and so on. The *arcs on demand* scheme (Figure 4) overlays arcs representing causal connections to or from a selected cell in a spreadsheet display. The *special forms* scheme (Figure 5) modified the graph representation to include special spreadsheet-like groupings of nodes to represent common patterns of connections among factors, such as sums or products. Arcs connecting nodes within these special groupings would be omitted, and the nodes would be packed closely together, saving considerable screen space.

In attacking the problem of packing more descriptive information into the representations we considered only solutions in which added information could be presented *simultaneously* for as much as the model as was in view. Our analysis and observations persuaded us that selective display of such information, which the user would have to request explicitly for a small part of the model, would not be as useful. If added information is needed when scanning the model as a whole, as when one is looking for a factor related to some real-world issue, one cannot afford to request information bit-by-bit to guide the scan. This limitation eliminated a number of otherwise useful techniques, such as fish-eye viewing [3], which augment the view of only a part of a structure at a time.

We devised two ways to present expanded descriptive information for large parts of a model simultaneously. In the *tiling* scheme (Figure 7), applicable only to a graph representation, the user could request a view of the model in which the nodes representing factors have been expanded in place as much as possible without overlapping. Added descriptive information is placed in the expanded nodes. This mode of displaying the model obscures most of the arcs but does not change the relative position of the nodes. We conjecture that users can maintain some sense of the overall structure of the model while viewing the expanded descriptive information, because of the invariance of the position of the nodes.

In the second scheme, *stairways* (Figure 6), cells or nodes are embedded in a three dimensional space. While the scheme can be applied to graphs or spreadsheets, it is easiest to visualize in the spreadsheet case. Imagine painting a column of spreadsheet cells onto the fronts of the steps of a flight of stairs. Viewing the stairs head on, the cells appear to be stacked above each other as usual. But viewed from a higher vantage point the tops of the steps become visible, and added information about each cell could be placed there. Three dimensional graphics could be used to produce these views of spreadsheet cells under user control.

In developing these ideas we did not explore a family of approaches that might be called *abstraction* schemes, such as representing parts of a graph by single nodes. These techniques clearly have potential for improving screen space usage. Unfortunately, we simply have not yet had time to include them in our analysis.

Before undertaking reanalysis of our user tasks with these new representation schemes or combinations of them, we constructed a preliminary realization of each one. For example, we were able to mock up the coloring scheme quite easily in Excel, and we pasted up a stairway from index cards and pieces of a printed spreadsheet. These crude efforts proved quite useful in elaborating our preliminary design ideas, and in seeing limitations in them. For example, the arcs on demand scheme seems unattractive for cells stacked in columns (a common arrangement in spreadsheets) because arcs connecting cells in the column tend to merge together. The stairway scheme when applied to a spreadsheet provides space that is quite narrow, since it is only the width of a cell, but some information, including formulae, fits better in a wider space.

We also discovered at this stage that the special forms scheme, conceived as a way of making a graph representation more compact, had potential advantages within a spreadsheet representation. In this scheme nodes arranged in specific ways are linked computationally in specific ways. The special arrangements of cells would be marked to indicate whether a total, product, or quotient (say) is being depicted. Thus by looking at the arrangement of nodes one can tell what computation is involved. In the example in Figure 5 one sees a number of quantities and their totals, and this arrangement of nodes would always have this interpretation.

This idea solves an important problem with spreadsheets. Whatever the arrangement of cells in a spreadsheet, the computation is always specified in the formula, which cannot be seen in normal viewing. It is thus common for spreadsheets to contain undetected errors, because the arrangement of cells can look correct while the actual computation is not what is needed. Applying the special forms scheme to a spreadsheet representation, this problem could be eliminated, at least in common cases, because the computation would be specified by the visible arrangement of cells, not by invisible formulae. A column of cells like that in Figure 5 would always compute a total. If the cells in question were not supposed to form a total, the fact that they were arranged in a column would signify an error to the user. If other cells were supposed to take part in the summation that were not in fact in the column, this too would signal an error to the viewer. Other special forms would visibly specify products, quotients and other computations without reliance on invisible formulae.

After pondering the preliminary realizations we identified three combinations of these schemes for further consideration, dropping the arcs on demand scheme from consideration due to the problems discovered earlier. The final candidates were a spreadsheet representation with coloring, special forms, and stairways; a graph representation with special forms and tiling; and a graph model with special forms and stairways.

**Reanalyzing tasks as performed with the new representations.**
With these three provisional representations, we revisited our task analysis to assure ourselves we had at least hit most of the trouble spots for the provisional representations. The reanalysis proved illuminating, bringing out potential problems we had missed in the design phase. In particular, the task analysis pointed out possible problems with the coloring scheme: if a colored cell is off the screen some special indication would be needed to point out its existence (this problem does not arise for arcs since the arc can be seen leading off the screen).

Thinking through the application of the stairways idea to large models raised an issue there, as well. The tops of the stairs would necessarily be deeper than the front faces, in order to fit more information than just the value on the top face. Tilting such a staircase may cause cells near the bottom and top of the screen to rotate out of sight, off the screen. This may cause users difficulty, in that they can not view the same "amount" of the spreadsheet in both views – information that was previously accessible in one screen is now spread over two.

Apart from these issues, we were satisfied that the three designs were at least feasible and worthy of additional consideration.

**Collecting User Data as Check on Analysis.**
We have not completed user testing on the three designs. The stairway designs are difficult even to mock up effectively with current tools. We have made a prototype of a graph model with tiling in HyperCard, and have made observations of three test users working with it. The prototype does not as yet include special forms. The tiles give more information about each node, in particular information about how the value of that node is encoded if it is not particularly self-evident to the viewer, and what an increase or decrease in that nodes value relates to in the real world.

The result of this user testing was mixed. While users found the extra information in the tiles useful, and had no trouble relating it to the model only one user found and viewed the information without at least some prompting from the experimenter. Thus the tiling scheme, while providing the necessary information, does not solve the problem of making users aware of the availability, and usefulness, of that information. Further testing is required to determine whether a different implementation of the tiling scheme could avoid this problem.

## DISCUSSION

We cannot argue that any of the representational innovations we have described will prove effective. We do argue that they are of sufficient interest and novelty to justify interest in the method that produced them. We want to argue further that they are of interest beyond the context of the specific application to which they were addressed, because they raise quite general issues for designers of representations.

One of these issues is *the value of presentation in context*. What is gained when detailed information (causal relationships, in this example) is displayed embedded in the global context where it is used (i.e., the entire model), rather than being shown in isolation? We have a strong intuition that presentation in context is a good thing – it seems likely that the accidental discovery of additional relevant information, as described earlier, and the process of learning about a complex model in general are strengthened by presenting greater amounts of information in the context to which it is applied. Our search for representations was biased by this intuition, although we think further analysis of the issue is needed.

As we noted, one situation in which presentation in context appears necessary is in scanning for information, provided that the to-be-presented data is necessary in forming judgements about the degree of interest in that particular area of the representation. Allocating more screen space to information about particular items, so that a sought-for item can be readily identified, means that less of the overall representation can be seen at any one time. This in turn requires more navigation over the representation's total area during the scan, making access of the required information more difficult, especially if information once on screen is now off. This tradeoff should be amenable to analysis.

A second situation in which presentation in context seems valuable is what we call uploading. If information about causal relationships (in our application) is delivered out of context, on demand, it might be possible for users to answer any question they have by directly consulting the model. By accessing information out of context, however, they may not be able to form a workable representation of the model in their own minds, and hence be unable to reason about the model or use the model to inform their own thinking. This intuition does not seem as easy to test.

A second general issue raised by our design problem is *the value of exploiting people's ability to relate different views of three dimensional forms*. The stairway scheme was devised as a way to tie two views of a structure together in a way that viewers could easily comprehend. By making the two views, which present different but related information, actually two views of the same three dimensional object, we hope to establish a strong and simple connection between them. Similar reasoning can be found in Robertson's [11] discussion of natural scene representations, and Card, Mackinlay, and Robertson's [1,7,10] uses of three dimensional representations . The intuition is clearly in need of test, and, at least in our own application of it, is open to argument. We think it is possible that the stairway representation will fail in practice because people's ability to comprehend three dimensional objects may be *too* great. Having seen the tilted view of a stairway of cells, people may "see" only the stairway and lose the relationship among the cells. After all, the cells on the front face of the stairway don't actually form a column; they only look as if they do from one particular well-chosen angle.

An analogy can be drawn with views of a cube. From a certain angle a cube looks like a regular hexagon. But most people do not see this in viewing a cube, even if they know it is true. It depends upon willingly disregarding clues about the 3-dimensionality of the solid, and perceiving the 2-dimen-sional projection of that solid as a geometrical figure. In a similar fashion, the stairway scheme may depend on people willingly maintaining a special perspective of the representation, even after they see that it is a special case. A saving grace may be the fact that this is, after all, a computer-generated representation and not a solid object in the real world. As such, depth cues and the like can be manipulated (and even eliminated) to help maintain the illusion(s) intended by the designers.

Returning to our design method, we briefly note a few lessons from it. First, early protoypes and user observation are absolutely essential. While task analysis can provide us with important insights as the potential problems and pitfalls a representation may face, it doesn't tell the whole story. User testing provides important additional insights which we believe we would not have found through task-analysis alone. Prototypes also provide surprising insights into what will and will not work for a given representation.

Second, as a corollary to the first, work in this sphere is limited by the tools available to us. There is good news here, in that Excel and HyperCard were very serviceable in mocking up many of our ideas in a reasonable amount time. The work reported here took on the order of 150 person-hours spread over six calendar weeks – a fair amount of that in the analysis and user testing phases. The actual mock-ups were fairly straight forward to construct, although somewhat tedious. Many of the effects of "executing" aspects of the models were accomplished by deciding ahead of time how such an effect would appear and providing a HyperCard card that could be shown at the appropriate times. HyperCard's smooth transitions from card to card made these kinds of 2-d effects easy to obtain. The bad news is that ideas like the stairway require a lot more work to explore – as the same kinds of transitions in 3-d are not as easily available to representation designers.

We close with what we think is the main lesson. Analysis of tasks is a good way to promote innovation in representation design. But, as in interface design in general, task analysis must be integrated with early prototyping and user-testing to provide the most benefits possible.

**ACKNOWLEDGEMENTS**

We are grateful to US West Advanced Technologies for support for this research. We thank our US West collaborators Warren Kuehner, Mike Epstein, and Ed Freeman, and our University colleagues Jeff McWhirter and Tammy Sumner for helping to create the intellectual setting for this work, and our colleague John Rieman for taking the time to criticize a draft of this paper.

**REFERENCES**

1. Card, S. K., Robertson, G.G., & J.D. Mackinlay. The Information Visualizer: An Information Workspace. In Proceedings of CHI '91 (New Orleans, Louisiana).

2. Casner, S. A Task-Analytic Approach to the Automated Design of Graphic Presentations. ACM Transactions on Graphics, 10, 2 (April 1991), 111-151.

3. Furnas, G. Generalized Fisheye Views. In Proceedings of CHI '86 (Boston, MA).

4. Krantz, D.H., Luce, R.D., Suppes, P. and A. Tverksy. Foundations of Measurement (Vol 1). Academic Press, New York, NY, 1971.

5. Licklider, T. R. Ten Years of Rows and Columns. Byte, 14, 13 (Dec. 1989), 324-331

6. Mackinlay, J.D. Automating the design of graphical presentations of relational information. ACM Transactions on Graphics, 5(2 April 1986), 110-141.

7. Mackinlay, J.D, Robertson, G.G., & S.K. Card. The Perspective Wall: Detail and Context Smoothly Integrated. In Proceedings of CHI '91 (New Orleans, Louisiana).

8. Microsoft Corporation. Microsoft Excel 3.0, One Microsoft Way, Redmond, WA 98052-6399

9. Palisade Corporation. @Risk Risk Analysis Package. 31 Decker Rd., Newfield, NY 14867

10. Robertson, G.G., Mackinlay, J.D., & S.K. Card. The Perspective Wall: Detail and Context Smoothly Integrated. In Proceedings of CHI '91 (New Orleans, Louisiana).

11. Robertson, P.K. A Methodology for Scientific Data Visualization: Choosing Representations Based on a Natural Scene Paradigm. In Proceedings of Visualization '90 (San Fransisco, California).

12. Roth, S. F. & J. Mattis. Data Characterization for Intelligent Graphics Presentation. In Proceedings of CHI '90 (Seattle, Washington).

13. Wehrend, S. & C. Lewis. A Problem-oriented Classification of Visualization Techniques. In Proceedings of Visualization '90 (San Fransisco, California).