Subjunctive Interfaces: Extending Applications to Support Parallel Setup, Viewing and Control of Alternative Scenarios

ARAN LUNZER
Hokkaido University
and
KASPER HORNBÆK
University of Copenhagen

Many applications require exploration of alternative scenarios; most support it poorly. Subjunctive interfaces provide mechanisms for the parallel setup, viewing and control of scenarios, aiming to support users' thinking about and interaction with their choices. We illustrate how applications for information access, real-time simulation, and document design may be extended with these mechanisms. To investigate the usability of this form of extension, we compare a simple census browser against a version with a subjunctive interface. In the first of three studies, subjects reported higher satisfaction with the subjunctive interface, and relied less on interim marks on paper. No reduction in task completion time was found, however, mainly because some subjects encountered problems in setting up and controlling scenarios. At the end of a second, five-session study, users of a redesigned interface completed tasks 27% more quickly than with the simple interface. In the third study we examined how subjects reasoned about multiple-scenario setups in pursuing complex, open-ended data explorations. Our main observation was that subjects treated scenarios as information holders, using them creatively in various ways to facilitate task completion.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI), interaction styles, evaluation/methodology*

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Subjunctive interfaces, multiple views, information exploration, user study

The work reported in this article was carried out while A. Lunzer was employed by the University of Copenhagen.
Authors' addresses: A. Lunzer, Meme Media Laboratory, Hokkaido University, North 13 West 8, Sapporo 060-8628, Japan; email: aran@meme.hokudai.ac.jp; K. Hornbæk, Department of Computing, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark; email: kash@diku.dk.

ACM Reference Format:
Lunzer, A. and Hornbæk, K. 2008. Subjunctive interfaces: Extending applications to support parallel setup, viewing and control of alternative scenarios. ACM Trans. Comput.-Human Interact. 14, 4, Article 17 (January 2008), 44 pages. DOI = 10.1145/1314683.1314685 http://doi.acm.org/10.1145/1314683.1314685

1. INTRODUCTION

'One swallow does not make a summer.' This saying, found in at least English and Danish, warns against making extravagant inferences on the basis of a single instance. And just as the sighting of a single bird is insufficient as evidence of summer's arrival, a single result obtained using a computer application will often be insufficient to meet a user's information needs. In applications through which choices are made—be it of a product to buy, an itinerary for a trip, or the design for a new building—we often try alternative scenarios and compare their merits. Even when no explicit choice is required, multiple scenarios may help in revealing the range of possible results, such as when simulating the outcomes of a business plan under alternative market conditions. Especially for tasks that require nontrivial problem solving, and have no formal criteria for identifying good solutions, there is a need for what-if exploration and comparison among alternatives.

Most applications, however, provide poor support for handling alternative scenarios. Users of e-commerce sites may be able to interact with only one product at a time among a range they would consider buying; travellers arranging their holidays may have to specify a unique departure time rather than a range of possibilities; and architects wanting to compare alternative floor plans must undo and redo a series of interactions. Thus, users interact through what has been called the single-state document model [Terry and Mynatt 2002a], which makes working with alternative scenarios laborious and cognitively demanding. Even applications that do present alternatives simultaneously (e.g., using a visualization with multiple coordinated views) tend to be limited in flexibility, and specialized to one application domain. Given that the need to work with alternatives is so widespread, mechanisms to support such interaction should be available generally, much as undo facilities have come to be expected as an unobtrusive part of today's applications.

Subjunctive interfaces are a way to provide such mechanisms by extending applications to support parallel setup, viewing and control of alternative scenarios. The concept of a subjunctive interface was inspired by Hofstadter's [1979] playful notion of a subjunc-TV—a magical television whose tuning knobs would provide access to alternative versions of a given broadcast. This article describes the principles underlying subjunctive interfaces, and presents example applications for information access, real-time simulation, and document design. In addition, we present three studies of a particular instance of a subjunctive interface for information access. Through these studies we investigate the usability of the interface, and conduct a detailed analysis of users' interaction with it. Our aim is to strengthen the empirical literature on subjunctive interfaces through usability-driven iterative development, and to highlight unresolved issues in the design and understanding of multi-scenario interfaces in general.

In Section 2, we present examples of subjunctive interface for common styles of application, and describe the key principles underlying those examples. We also expand our argument on the state of current applications, and discuss related work. Sections 3 to 8 present the empirical studies and their possible interpretations. Finally, in Section 9, we integrate the results of the studies and discuss future work.

2. SUBJUNCTIVE INTERFACES

Subjunctive interfaces support users in setting up and working with many scenarios in parallel. In this section we present three examples showing how applications can be extended with a subjunctive interface; we then explain the principles embodied by these examples.

2.1 Examples of Subjunctive Interface

The three examples in this section—a census-data browser, a simulation of ant behaviour, and a document editor—can be seen as instances of common application styles: information access, real-time simulation, and document design. For each example we describe difficulties faced by users who wish to explore alternative scenarios, and show how a subjunctive interface can alleviate these difficulties. The examples are not intended to be fully functional applications; rather we wish to illustrate how commonly used applications may benefit from a subjunctive interface.

2.1.1 *A Census-Data Browser.* The upper part of Figure 1 shows a browser, based on Hochheiser and Shneiderman's [2000] simultaneous-menus design, for accessing a database of 828 records on commercial activity in the state of Maryland. Following Hochheiser and Shneiderman's terminology, we refer to the clickable lists of counties, industries and years as menus. Once the user has selected an item in each menu, the system displays the corresponding record's statistics for employees, payroll and establishments. The selection in a menu can be changed simply by clicking on a different item, thereby accessing a different record.

This application provides good support for requesting any individual record, but not for tasks that involve accessing a large number of records. For example, a user who wishes to scan all the data for some county (i.e., all the industries over all the years) must select each combination of industry and year in turn. This is not only burdensome in terms of the sheer number of mouse clicks needed, but also requires the user to organise the iteration so that all desired combinations are covered.

The lower part of Figure 1 shows a subjunctive interface for this application. This interface can support multiple scenarios, where a scenario in this application comprises a selection in each menu, and the corresponding statistics. The overall layout of the browser's elements (i.e., the menus and the three result displays) is unchanged, but each menu can support distinct selections in

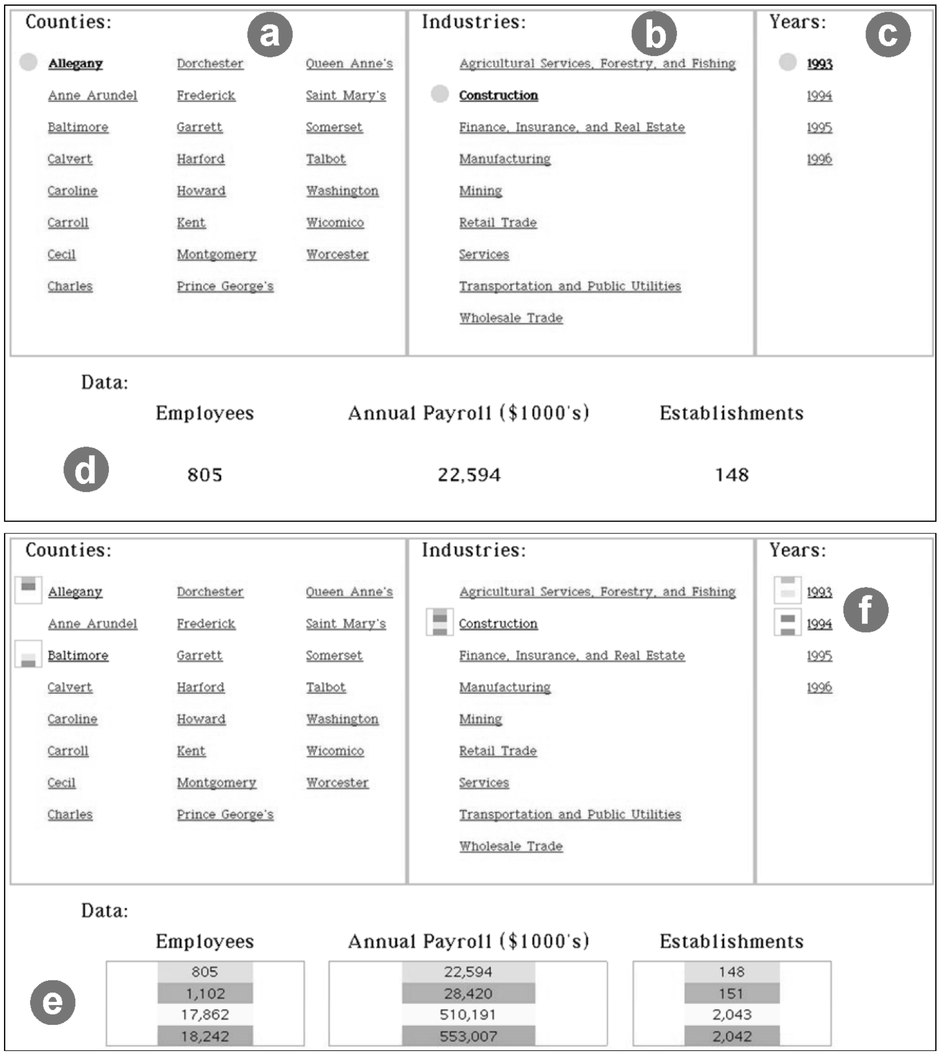


Fig. 1. A simple and a subjunctive interface for browsing census data. Above: A simple browser. For a selected county (a), industry (b), and year (c), the results area (d) shows the number of employees, total annual payroll, and number of establishments. Below: A subjunctive-interface version of the browser, supporting multiple scenarios. The user has set up four scenarios holding the Construction statistics for both Allegany and Baltimore, in 1993 and 1994. Correspondence between menu selections and result values is indicated with position and color cues in the result displays (e) and the markers next to menu items (for example, f). The values 805, 22594 and 148 at the top of the result displays, for example, relate to Allegany in 1993.

the various scenarios, and each result display shows a separate value for each scenario. The setup in Figure 1 has four scenarios.

The subjunctive interface supports inter-scenario comparison. For example, a user who wants to rank a county's industries according to some statistic can set up all the industries' statistics side by side and simply read off their respective values. By contrast, with the original browser (which we refer to as the simple interface, because it supports just one scenario) the user would have to select each industry in turn, and to remember or write down its results for comparison with the others. The ease of setting up such comparisons also contrasts with the complex queries that users would have to master if the data were offered instead through an SQL-style interface.

When the subjunctive interface is first opened, or after being reset, it behaves just like the simple interface: selections made by clicking on menu items cause the corresponding record to be displayed. However, the menus in this interface also support operations for setting up and working with multiple scenarios side by side. New scenarios are set up by a menu operation that copies existing scenarios; once this is done, the operations for adjusting a single scenario can be used to adjust many scenarios simultaneously. For the scanning task described above, a user could set up parallel scenarios showing the statistics for all the industries in one year, then move these scenarios through successive years by making a single click on each year in turn.

This example suggests that being able to handle multiple scenarios in parallel may increase the efficiency of working through and comparing alternatives. This style of application has been the focus of most of our work. To date, we have reported the census-data browser's preliminary designs [Lunzer and Hornbæk 2003], and the results of Studies #1 and #2 [Lunzer and Hornbæk 2004] that are summarised in Sections 3 to 6 of the present paper. Study #3, described in Sections 7 and 8, has not previously been reported. In the C3W prototype [Fujima et al. 2004] we applied a similar approach to information access through Web applications; Lunzer [2004] discussed the potential benefits of C3W and other subjunctive interfaces for information access.

2.1.2 *A Real-Time Simulation of Ant Behavior.* Figure 2 shows an interface for a simulation of ants' cooperative food-foraging behaviour, originally presented in Lunzer and Hornbæk [2003]. Parameters affecting the model are specified using sliders, and changes to the parameters are reflected immediately in the running simulation. Despite the immediate response, it is hard for a user to judge how much of what happens next is as a result of the adjustment. If a strong pheromone path is seen to disperse shortly after the user increases the diffusion rate, for example, it would be helpful to see how the path would have appeared without that increase. Such comparisons are easier if the alternative cases can be seen side by side, which becomes straightforward when the application is extended with a subjunctive interface as shown on the right side of the figure.

In the subjunctive-interface version of the simulation, the parameter sliders support the creation and adjustment of multiple scenarios. In the figure shown here the user has set up three scenarios, by manipulating the diffusion-rate and evaporation-rate sliders. To allow multiple scenarios to be seen side by side without using large amounts of screen space, each scenario's simulation display is shrunk; alternatively, the display can be pinned so that just one scenario at a time is shown at full size.

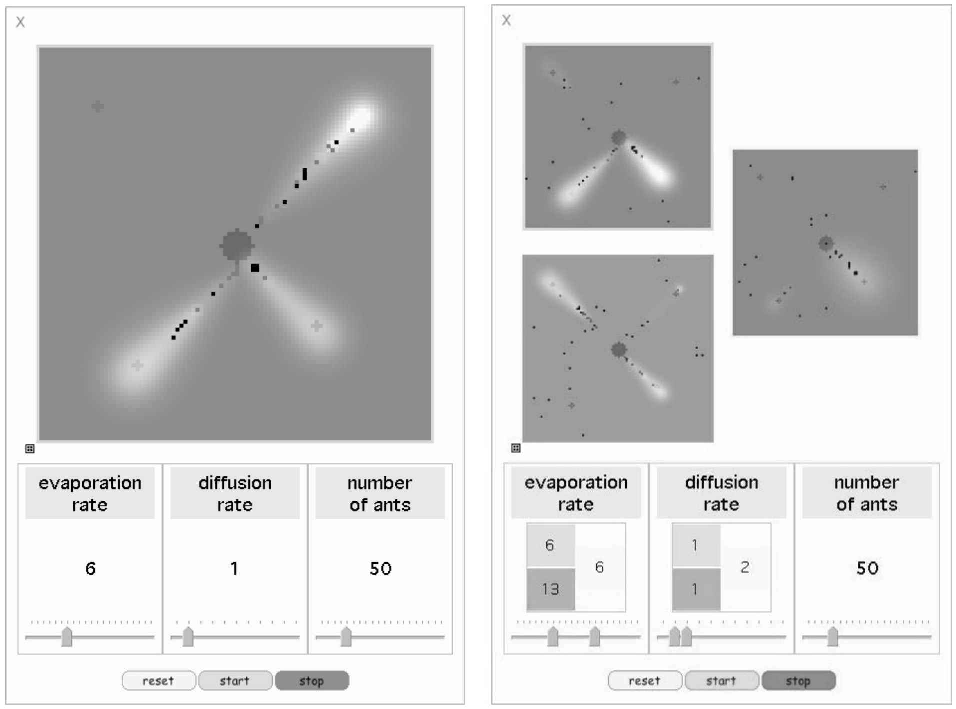


Fig. 2. Simple and subjunctive-interface versions of a simulation of ants' food-foraging behavior. Left: The simple interface. Ants (black and red dots) are seen searching for static food sources (red crosses). When an ant finds a food source, it picks up some food and heads home to the nest in the centre of the view, dropping pheromone (bright yellow) to help other ants find that source. As more ants follow the path, the pheromone builds up, but it is subject to natural phenomena of diffusion and evaporation at rates set by sliders underneath the simulation view. Right: A subjunctive interface for this simulation. The user has established three scenarios, combining various values for evaporation and diffusion rates. The three corresponding instances of the simulation are seen side by side. Like in the census browser, correlation of position and color is used to indicate which values belong with which results—for example, the simulation running at top left corresponds to an evaporation rate of 6 and diffusion rate of 1. Notice that because all scenarios have the same value for number of ants, only a single view of this parameter is provided.

This interface, which has previously been reported as an example of using subjunctive interfaces for e-learning [Jantke et al. 2005], suggests that being able to see scenarios side by side can bring benefits to applications where the outcome evolves over time. Comparing scenarios in such applications is usually difficult, since the evolution of the outcome is hard either to remember or to record; running scenarios in parallel lets the user observe the differences as they occur.

2.1.3 *A Document Editor.* A common challenge in creating a document is finding a layout policy that results in an attractive layout. In an interface supporting only a single scenario, a user may select some policy (e.g., specifying that figures are embedded inline), then try to adjust the document's text and figures so that the overall layout is acceptable. Being able to set up multiple

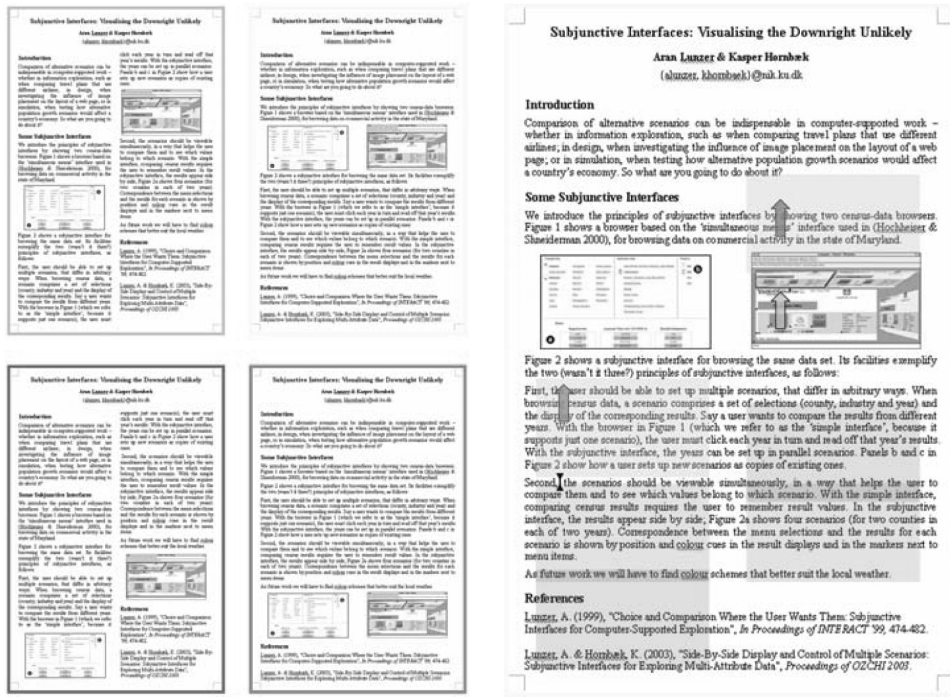


Fig. 3. A subjunctive interface for a document editor. A user is working with four possible designs for a page of text with embedded figures. The designs all involve the same text and figures, but with two policies for figure placement and a choice of single- or two-column layout. On the left, all the designs are seen as thumbnails. On the right, the user has pinned the multiplexer to full size and is working with the version from the upper right (yellow) scenario. In this situation, the user can request the overlay of information relating to the other scenarios; translucent colored blocks show the alternative positions of the figures, while colored arrows show where the current text-insertion point would appear in each case.

scenarios that reflect different policies, then to adjust the scenarios in parallel, could speed up the process of finding a good layout.

Figure 3 shows how a document editor with a subjunctive interface could support such exploration. By manipulating the controls that set single- or two-column layout, and the policy to be used in placing figures, the user has set up four scenarios. These scenarios contain the same text, which can thus be updated in all scenarios simultaneously. The left of the figure shows the four scenarios shrunk to fit the area normally taken up by the single page layout. This makes it easy to compare the overall placement of the paragraphs, for example, but these reduced views may be too small to support editing. On the right of the figure, the user has pinned the display so that just one scenario is shown, at full size. This makes it easier to edit the text, but harder to track the effects of this editing in the other, unseen scenarios. The interface can assist by overlaying cues that reflect selected properties of those scenarios; in this figure the user sees semi-transparent blocks and markers that show, respectively, the positions of the figures and of the text cursor within the three other scenarios.

Census browser for Study #1	Census browser for Study #2	Ant simulation
(a) Clicking and holding on the menu item that the new scenarios are to include generates a pop-up; choosing the copy icon copies the active scenarios, or all scenarios if Alt key is simultaneously pressed.	(b) Control-clicking and dragging a menu marker, then dropping it on a menu item, copies the scenarios in the dragged marker and sets the target item's value in the new scenarios.	(c) Control-clicking a slider thumb copies the scenarios corresponding to that thumb, and creates a new thumb that can be dragged to another value.

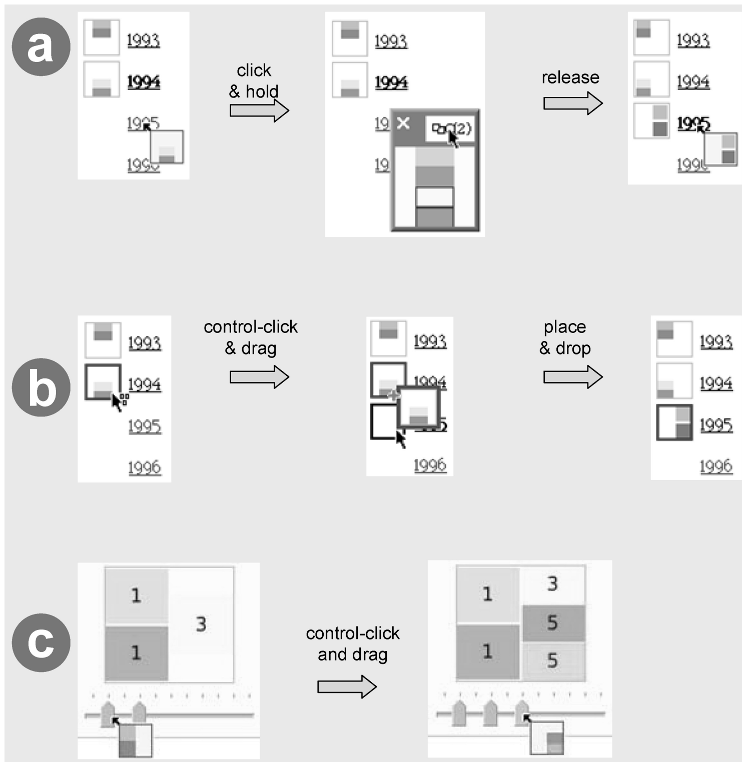


Fig. 4. Creation: User setup of one or more extra scenarios.

This example, though still at the mock-up stage, suggests that subjunctive interfaces are suitable not just for applications in which the input choices (e.g., menu settings) only affect the outcome indirectly, but also where some inputs—in this case, the text and figures—are an integral part of the outcome.

2.2 Design Principles for Subjunctive Interfaces

The examples above illustrate three key design principles for subjunctive interfaces: the interface should support (1) setting up multiple independent scenarios that exist at the same time; (2) viewing those scenarios side by side; and (3) making changes to many scenarios in parallel. Below, we motivate these principles and discuss how the necessary facilities can be realized, with the help of Figures 4 to 9 where we contrast the facilities as implemented in the census-data browser used for Study #1, a redesigned browser used in Study #2, and the ant simulation.

2.2.1 *Setting Up Multiple Independent Scenarios.* The key property of an application with a subjunctive interface is that it can support multiple, mutually independent scenarios at the same time. The user has simultaneous access to these scenarios, which the application maintains as independent computations.

We expect that in most applications the main reason for a user to create multiple scenarios will be to explore the impact of alternative input values (though the ant simulation, with its stochastic behavior, is a case where observing multiple scenarios that have identical inputs may also be of interest). In the subjunctive interfaces built to date, we have therefore made the creation of multiple scenarios an operation supported by the input widgets. Figure 4 shows how a menu in the census browser and a slider in the ant simulation allow the user to copy one or more existing scenarios, and to specify the new scenarios' setting for that input (for all other inputs, the new scenarios will have the same values as the scenarios that were copied). When the operation is completed, all input widgets and result displays are updated to include the new scenarios. In the ant simulation and the revised census browser, copying is initiated by interacting with the scenarios that are to be copied, whereas in the earlier census browser it was by interacting with the desired new value, with the scenarios to be copied being determined by an "active scenarios" setting. This setting, discussed in detail below, refers to the designation of some scenarios as being the default target for a subsequent operation.

A subjunctive interface must also provide operations for deleting scenarios; we show examples in Figure 5. However, given that the inputs of each scenario are independently adjustable, we expect users to explore alternatives not by repeatedly creating and deleting scenarios, but by adjusting their input values. Thus, deletion is chiefly a way to simplify the working environment, for example by discarding all but the most interesting scenarios found in some exploration. Nonetheless, if the creation and deletion operations are sufficiently lightweight, some users may decide to adopt a use-once approach to scenarios.

Creating multiple copies of an entire computation clearly involves some cost, both in terms of resources and interface complexity; we now consider the circumstances under which this cost is or is not compensated by benefits. First, independent scenarios are typically not needed if the inputs being varied have a trivially predictable effect on the application's outcome. For example, a graphic designer may want to try a range of formats for some text label; if these formats do not influence the rest of the workpiece (such as causing other elements to be rearranged) then the designer would gain little by working with multiple copies of the entire design. Exploring formats for the individual label may be supported more effectively by a technique such as Side Views ([Terry and Mynatt 2002b], discussed in Section 2.3).

Second, there is no need to create scenarios based on exhaustive combinations of alternative values for inputs that do not affect each other—such as a range of font styles on the one hand, and a range of image-quality settings for

Census browser for Study #1	Census browser for Study #2	Ant simulation
Pressing the delete key when mouse is over a menu marker deletes the scenarios in that marker. Similar to (a). (not shown) Pressing the delete key when mouse is over a result display deletes the active scenarios. In either case, if no scenarios would remain the operation is rejected.	(a) Pressing the delete key when mouse is over a menu marker deletes the scenarios in that marker. (b) Pressing the delete key when mouse is over a result display deletes the last (bottom-right) scenario. In either case, if no scenarios would remain the user is asked to confirm reset of all menus.	(c) Pressing the delete key when mouse is over a slider thumb deletes the scenarios corresponding to that thumb. Pressing the delete key when mouse is over a slider-value display or the simulation area deletes the last (bottom-right) scenario. Similar to (b). In either case, if no scenarios would remain the operation is rejected.

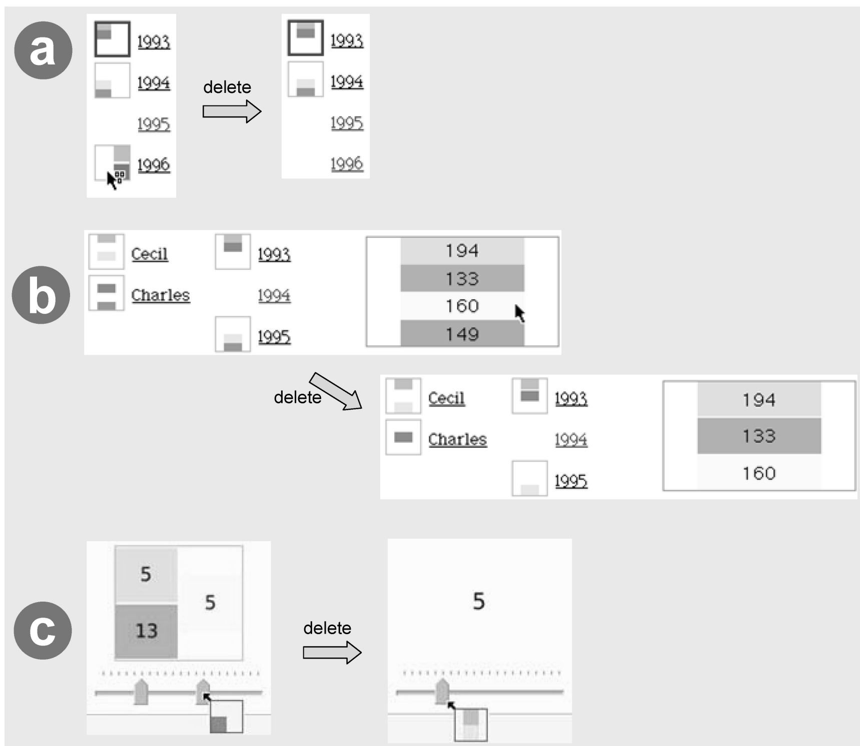


Fig. 5. Deletion: User deletion of one or more scenarios.

embedded pictures on the other. In such a situation, it would be more efficient for the user to explore the impact of a single input at a time. Conversely, sometimes the setting of one input may constrain the valid settings for others. If a document includes a global switch for color or black-and-white presentation, say, there would be no point in creating scenarios that combine the black-and-white setting with detailed color settings for individual elements. Outside these extreme cases, typically a user may wish to explore combinations of alternative values for several inputs, and should be supported in setting up scenarios with the desired value combinations. A user who is considering n font sizes and m margin widths for a document may want to try all $n \times m$ outcomes, or may be interested in only a subset of these outcomes (e.g., having the narrowest margins only with the largest font). Our example applications allow the user to set up scenarios with arbitrary input combinations, with the proviso that, due to issues of display resolution and complexity, there is a fixed limit to the number of scenarios that can exist at one time. This is related to the second principle.

2.2.2 *Viewing Scenarios Side by Side.* The second subjunctive-interface principle is that the user must be able to see scenarios side by side. This principle is motivated by our wish to support users in making comparisons between scenarios, and our belief that side-by-side presentation is an effective approach to providing such support.

Multi-scenario presentation should help the user to understand how scenarios differ, in terms of their inputs and outcomes, without undue burden on memory or need for external notes. However, the interface should not be so specialised to the highlighting of inter-scenario differences that viewing a single scenario becomes problematic.

Our earliest subjunctive-interface demonstrations [Lunzer 1999] addressed applications where the displays of inputs and outcomes tend to differ greatly between scenarios. In such cases, side-by-side viewing of multiple scenarios can be achieved simply by overlaying the scenarios' respective displays, perhaps adding some visual encoding such as color to help the user distinguish them. This approach can also work for low-density displays such as wire-frame graphical objects or line-graph plots.

However, for displays made up of text or dense graphics such visual overlay would result in an unreadable mess. One alternative would be to render multiple adjacent copies of the application's entire interface, but that wastes screen space and may make it difficult for a user to figure out which elements differ between scenarios—reminiscent of children's spot-the-difference puzzles. Our current approach, seen in the example applications, is based on *widget multiplexers*. A widget multiplexer is a user-interface element that handles the presentation and user interaction for some defined region, typically a single input widget or result display. If that region would appear differently in the scenarios that have been set up, the multiplexer shows all those appearances. Depending on the region being multiplexed, the presentation of multiple scenarios can differ: the census browser includes one type of multiplexer for its statistics displays and another for its menus, while the ant simulation has a general graphical multiplexer for the main display and specialised multiplexers for its input sliders. All types of multiplexer use a common set of colors and spatial layouts, to help users find all the values that correspond to a given scenario.

As shown in Figure 6, the census browser for Study #1 displayed a single, nonmultiplexed result if every scenario happened to have the same value, whereas in the browser for Study #2, the result displays show separate

Census browser for Study #1	Census browser for Study #2	Ant simulation
(a) Interface details showing the correlation of color and placement in menu markers and result displays. If a result has different values in different scenarios, the result display shows all scenarios' values side by side. But if a result value is the same in all scenarios, a single display serves for all scenarios.	(b) Each result display shows a separate value for each scenario, whether they differ or not.	(c) For the sliders: • If the slider has different values in different scenarios, each value has its own slider thumb. The values are shown side by side. • If the slider has the same value in every scenario (in this figure, the number of ants), just one slider and one value are shown. For the simulation, a separate view is shown for each scenario.

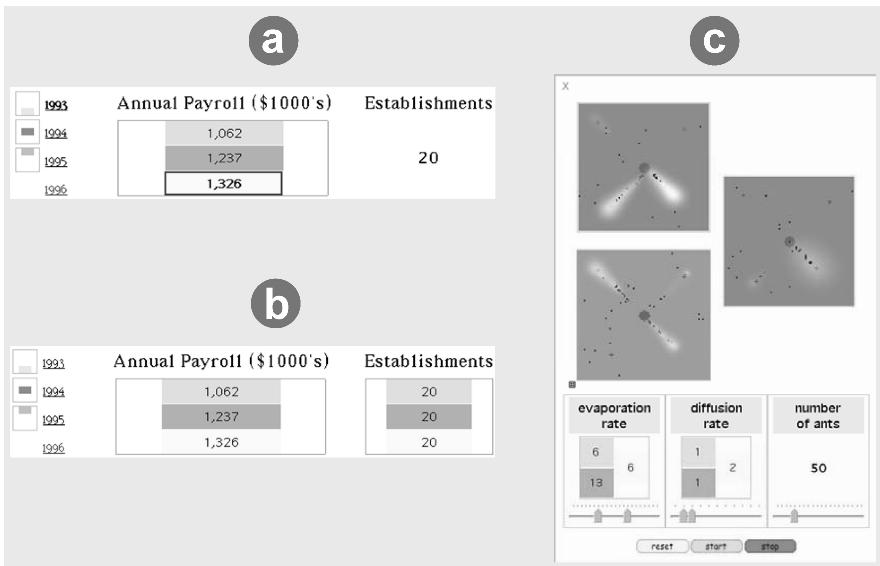


Fig. 6. Side-by-side display: Facilities for displaying multiple scenarios.

values for all scenarios even if they are identical. This design change was made after observing that subjects in Study #1 were sometimes confused by seeing just a single result in the presence of multiple scenarios. On the other hand, Figure 6 also shows that the sliders in the ant simulation still display only a single value if all scenarios have that value. For displaying inputs, we believe that this behavior is preferable.

Figure 7 shows facilities that let the user alter the display to make particular scenarios or their relationships more prominent. Again there is a difference between the two versions of the census browser. As one aspect of its handling of active scenarios, the earlier version highlighted one scenario by showing its

Census browser for Study #1	Census browser for Study #2	Ant simulation
(a) A global setting highlights the menu-item labels and result values for one scenario. • Clicking on a value in a result display switches the highlight to that value's scenario. • When the active-scenarios setting changes, the highlight setting will change automatically if the previous highlight scenario is no longer one of the active scenarios.	(b) No highlight setting, but scenarios can be rearranged: clicking and dragging on a scenario's value in a result display makes that value follow the mouse within the display, with other scenarios' values moving to make room.	Scenarios can be rearranged, by clicking and dragging on a scenario's value in a slider display as in (b). (c) Shift-clicking and dragging on a slider-value display enlarges the displays of the scenario under the mouse and shrinks the other scenarios. (d) Clicking the pin control in the simulation display switches it to and from full-size display mode, in which scenarios are shown on separate selectable tabs.

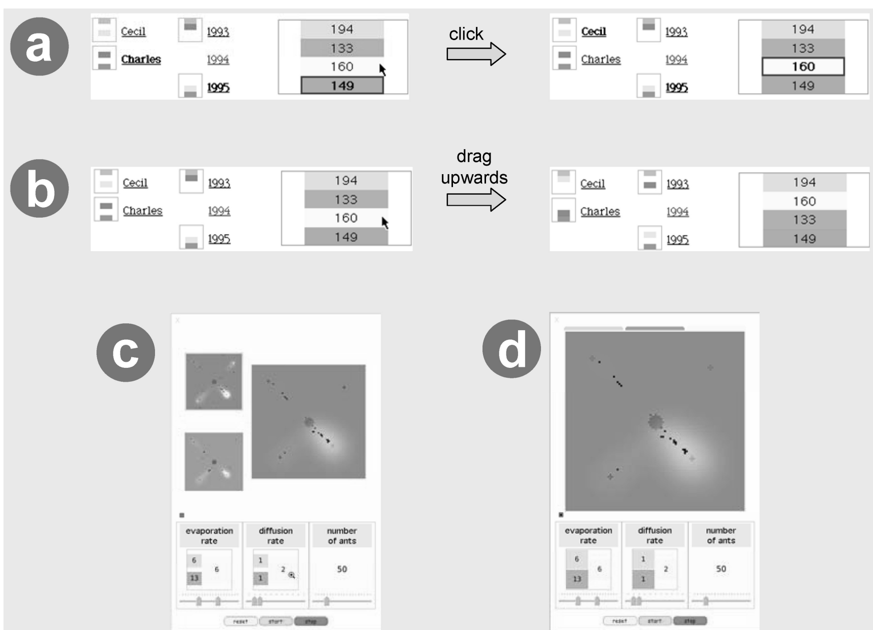


Fig. 7. Display configuration: User configuration of scenarios' displays, for example, to increase salience of particular values or comparisons.

selected menu items in bold text and its results with a black frame. The user could change which scenarios were active, and hence move this highlighting, by clicking on the result displays. The later browser, which does not deal with active scenarios in the same way, does not provide this highlighting but does let the user click and drag result displays to change how scenarios are arranged.

The use of such rearrangement in Study #3, in particular, will be discussed in the findings of that study. The ant simulation, as well as supporting scenario rearrangement, provides facilities for recovering detail lost by shrinking multiple results to fit into the screen space normally used for one: the user can temporarily enlarge one scenario's displays at the expense of the others, or can pin a multiplexer to a view in which results are displayed at full size on separate tabs. The document editor example (Figure 3) embodies a hybrid approach that combines multiplexing and overlay; in the pinned presentation, one scenario is shown in full detail but is overlaid with unobtrusive indications of where salient elements are located within the other scenarios.

In general, creating a subjunctive interface for an application involves choosing multiplexers suited to the kind of inter-scenario variation to be supported. The multiplexed menus used in the census browser can support scenarios that differ in terms of which item is selected, but would not suit an application in which different scenarios could have different items on the menus. Furthermore, an application in which a choice on one menu determines whether or not some second menu is available would require a different type of multiplexer again.

A key question in the design of subjunctive interfaces is how many scenarios to support simultaneously. In our example applications, the widget multiplexers are designed to show up to twelve scenarios side by side; our description of the results of Study #3 includes some observations on how users responded to this limit.

2.2.3 *Making Changes to Many Scenarios in Parallel.* When several scenarios have been set up, the user's interface operations can be applied to many scenarios at the same time. The motivation for this principle is to improve exploration efficiency, in terms of the number of operations required to examine some desired range of scenarios. For example, with the right setup of scenarios in the census browser, a user can scan all combinations of industry and year just by clicking on each year in turn; in the document editor, edits to the text can be applied to all scenarios simultaneously.

To support this principle, a subjunctive interface must provide control over which scenarios will be affected by an operation. In the interests of efficiency, the user should not have to specify this for each operation individually, so our interfaces typically support a persistent active-scenarios setting, as illustrated in Figure 8. In the census browser for Study #1, there was a global active-scenarios setting that reflected the target of the most recent operation, the intention being that a user could efficiently apply a sequence of operations to the same scenarios. Operations such as the creation of new scenarios therefore updated the setting automatically. The user could also change the setting explicitly, selecting or de-selecting scenarios by clicking on values within a result display, and could override the setting temporarily to affect all scenarios at once by holding down the Alt key while performing an operation. For reasons reported below in the discussion of Study #1, we abandoned this approach for the census browser: the browser for Study #2 supports a separate active-scenarios setting for each menu, determining which scenarios are affected by

Census browser for Study #1	Census browser for Study #2	Ant simulation
(a) A global setting, indicated by bold text in menus and results, and by the mouse pointer when over menu items. • User can change the setting by clicking on menu markers or result values, or implicitly by selecting a target scenario in an adjustment operation • Newly created scenarios automatically become active, as shown in Figure 4(a).	(b) A per-menu setting, shown by one marker in the menu being highlighted with a dark blue border. • User can change the setting by manipulating, or just clicking, a menu marker. • In a menu that has no selections yet, all scenarios are active.	No setting that affects sliders (c) A global setting for button operations (start/stop/reset). When the mouse is over a button, after a small delay a trigger appears. Moving the mouse onto this trigger creates a pop-up within which the user can change the setting by selecting or de-selecting scenarios. Below, the user has de-selected the blue (bottom left) scenario.

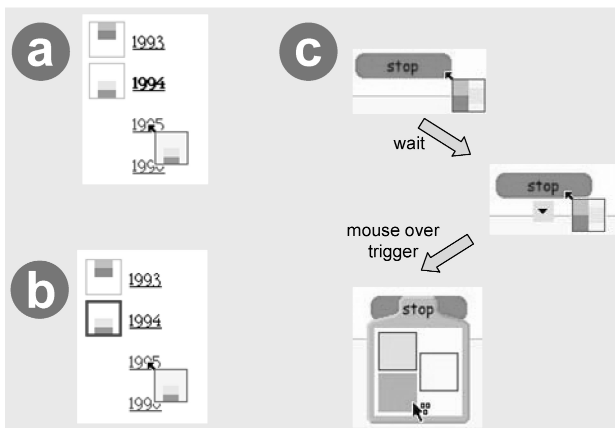


Fig. 8. Active scenarios: A setting that designates one or more scenarios as the default target of a subsequent operation.

the next click in that menu. However, as shown in Figure 8, we still believe that a global setting is appropriate for controls such as the buttons in the ant simulation.

Figure 9 shows the mechanisms for adjusting the contents of one or more scenarios. Clicking on a menu item in either version of the census browser adjusts the active scenarios, however defined. Additionally, in the census browser for Study #2 dragging a menu marker updates the group of scenarios indicated by that marker, dynamically, just as dragging a slider thumb in the ant simulation updates the group of scenarios currently set to that thumb's value. To adjust scenarios independently of their current groupings, in the census browser the user can choose a scenario within a pop-up selector on the new value wanted for that scenario, while in the simulation the user chooses scenarios within a slider thumb's pop-up before starting a drag. Our current impression is that the latter interaction style has greater flexibility.

Census browser for Study #1	Census browser for Study #2	Ant simulation
Clicking on a menu item sets that value in the active scenarios, or all scenarios if Alt key is pressed. Similar to (a). Clicking and holding on a menu item generates a pop-up; releasing mouse over a scenario sets the clicked value in that scenario. Similar to (b).	(a) Clicking on a menu item sets that value in the active scenarios for that menu. (b) Clicking and holding on a menu item generates a pop-up; releasing mouse over a scenario sets the clicked value in that scenario. (not shown) Clicking and dragging a menu marker dynamically adjusts its scenarios to hold the value that the mouse is over, finally taking the value of the item on which the marker is dropped.	For sliders: (c) Clicking and dragging a slider thumb dynamically adjusts the scenarios corresponding to that thumb. (d) When the mouse is over a thumb, after a small delay a trigger appears. Moving the mouse onto this trigger creates a pop-up. (e) Clicking on a blank scenario square in the pop-up sets the thumb's value in that scenario. (f) Selecting or de-selecting scenarios in the pop-up allows choice of scenarios to be affected by dragging or copying the thumb. For buttons: Clicking a button operates on only the active scenarios.

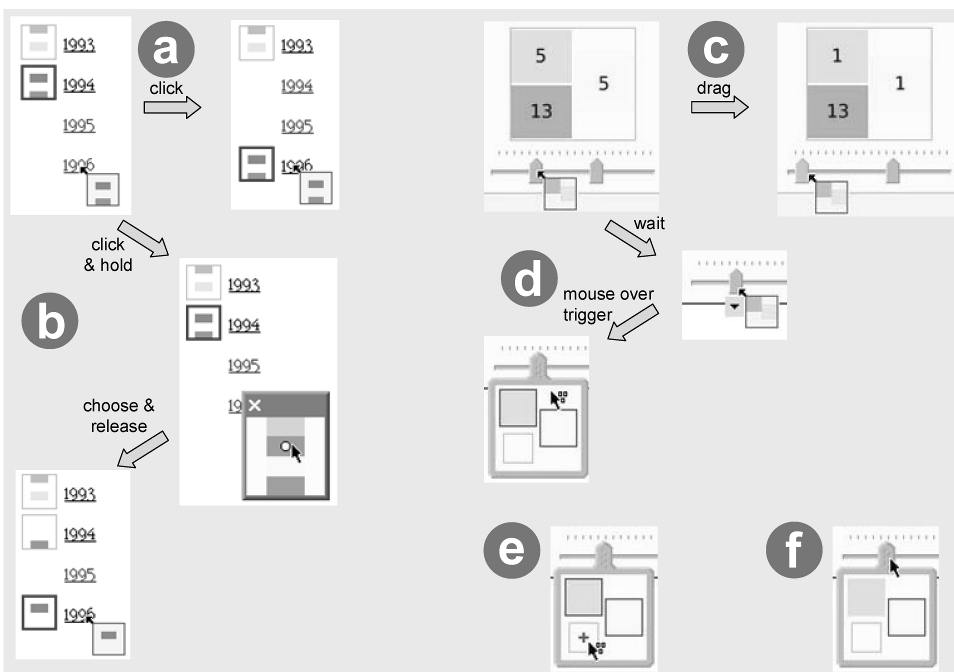


Fig. 9. Adjustment: Changing the contents of one or more scenarios.

This concludes our explanation of subjunctive-interface principles. In the next section, we examine how these principles, and our implementations embodying them, relate to other work on helping users to explore alternatives.

2.3 Related Work

The advantages of working with multiple scenarios in parallel have been argued from several perspectives. Lunzer [1999] described the advantages in terms of the cognitive dimensions framework, arguing that subjunctive interfaces improve visibility (the ease of bringing desired information into view), reduce viscosity (the overhead incurred in changing something that has been specified), improve juxtaposability (the ability to view information side-by-side, important in making comparisons), and reduce premature commitment (a feeling of being compelled to take a decision without sufficient information). Other perspectives focus on the characteristics of tasks facing users in many current applications [Terry and Mynatt 2002b; Lunzer and Hornbæk 2004]. Those tasks, it is argued, frequently require nontrivial problem solving and have no fixed route to their solution. Consequently, there is a need for what-if exploration of scenarios of interest and for interfaces supporting comparison of those scenarios. As an example, Terry et al. [2004] argued that studies of expert practitioners of website design and of image manipulation show how working with alternative solutions is crucial. A further example is given by Toomim [2006], who contends that duplication (and slight modification) of alternative scenarios is widespread in accounting, word processing, composing, and presenting. Another perspective is that of Tufte [1997, p. 105], who described the benefits of seeing alternatives side by side rather than one at a time. He introduced the notion of multiples, that is, the presentation of multiple images with similar formats next to each other, and argued that "Multiples directly depict comparisons, the essence of statistical thinking". Roberts [2000] likewise recommended view multiplicity in computer interfaces as a way to encourage users to try out alternatives.

Studies that begin to investigate these advantages include those by Terry and Mynatt [2002b] and Terry et al. [2004]. Terry has developed Side Views, a kind of previewing mechanism for exploring the effect of editing commands. A Side View shows a preview of applying a particular command on the active document. The user can interact with the Side View to vary the command it applies, and can choose to make the Side View persistent, so that it dynamically reflects changes to the document. Side Views can be applied to other Side Views, allowing the user to see the effect of a chain of commands. In addition to showing a single preview, a Side View can show how a range of parameters to a previewed editing command would affect the document. The benefit of being able to preview a range of outcomes before deciding which to use was also demonstrated in Pegasus [Igarashi et al. 1997], a prototype system for interactive beautification of geometric drawings. Because the range is generated automatically by the system, this interaction style is referred to as a 'suggestive interface' in follow-on work [Igarashi and Hughes 2001; Tsang et al. 2004]. Likewise, in Design Galleries [Marks et al. 1997] the system automatically generates and presents a range of graphical outcomes based on variation in input parameters, for example to help a user select the illumination for a rendered image.

While these preview interfaces all support a user in exploring alternative ways to proceed in a design task, at each stage progress depends on committing to one of the offered outcomes. Parallel Paths [Terry et al. 2004] removes this constraint, in that a user can choose several different versions of a command to be applied, thereby creating separate variations of the working document. These variations can then have further commands applied to them, with the user being able to control whether each command is applied just to one variation or to all simultaneously. Parallel Paths differs from the existing examples of subjunctive interface in that the technique for displaying variations (using so-called Parallel Pies) gives the user dynamic control over partitioning the workspace into regions, where each region is dedicated to one variation. This avoids having to make room to display multiple copies of the workspace, but provides a limited form of side-by-side comparison. Unlike in a subjunctive interface, the user can never see simultaneously how any given region of the workspace appears in the multiple variations. A second difference from subjunctive interfaces is that in the census browser (Section 2.1.1) and the ant simulation (Section 2.1.2) the interfaces include persistent, multi-scenario displays for each parameter (in addition to the displays for the results) so that the user can see how parameter choices are related to scenarios. By contrast, Parallel Paths shows only the document variants. There is no display to reveal how each variation has arisen, making Parallel Paths suited mainly to applications in which the end result captures all information of interest to the user.

More broadly, several existing types of application are related to the subjunctive interface principles. A full survey of such application types is outside the scope of this article, so here we introduce some examples and consider the degree of support that they provide. First, in information visualization the aim is to build interfaces that can show large amounts of information in a way that amplifies cognition. For viewing and comparing alternatives within tabular data, for example, interfaces such as Polaris [Stolte et al. 2002] provide rich facilities for constructing and reconfiguring the table. The Table Lens [Rao and Card 1994] lets a user visualize chosen rows relative to each other and to the full range of values in each column. These tools, however, are limited to data suitable for row-and-column display, while a subjunctive interface can be developed for more general forms of data. Tools such as the Influence Explorer [Tweedie et al. 1996] and Filmfinder [Ahlberg and Shneiderman 1994] project numerical or ordinal data onto a two-dimensional graphical layout, and provide interactive controls that allow a user to highlight data elements or ranges. Comparison is supported by the user's ability to switch the display rapidly and reversibly among different settings for the highlighting. Such dynamic switching is good for drawing attention to subtle distinctions, especially along some continuous range, but in other cases comparison may be better supported by simultaneous, side-by-side presentation of key cases.

Some styles of end user programming environment make it straightforward for users to set up side-by-side calculations, which could then be used to handle multiple scenarios. A survey of such programming environments is not relevant here, except for two styles that are directly related to subjunctive interface principles. The first is visual dataflow, in which a form of interactive wiring diagram is used to connect processing components; environments based on this approach include commercial visualization systems such as AVS and Iris Explorer, and scientific workflow frameworks such as Taverna [Oinn et al. 2006] and Kepler [Ludäscher et al. 2006]. A user's ability to create elegant multi-scenario interfaces in such environments largely depends on the existence of appropriate components, for example for drawing together many scenarios' results into a view that supports comparison.

The second style of end user programming environment that clearly supports side-by-side calculations is the spreadsheet, by virtue of its tabular structure and its built-in operations for duplicating calculations. For example, Chi et al. [1998] demonstrated the Information Visualization Spreadsheet, in which parallel rows can be set up to calculate distinct but related visualizations. Indeed, the principles listed by Chi et al. [1998] to explain the power of their spreadsheet can be seen as precursors to the subjunctive interface principles, in their emphasis on comparison, on viewing multiple features simultaneously, and on applying operations in parallel. Among the many systems whose design was inspired by spreadsheets, those demonstrated by Nardi [1993] and Levoy [1994] retain the tabular structure as a crucial feature, whereas those of Hudson [1994] and Burnett et al. [2001] dispense with this feature while emphasising others such as the programming model based on live, formula-based derivation. On the other hand, some systems dispense with the programming model while retaining the tabular structure, such as Jankun-Kelly and Ma's [2001] interface for exploring multidimensional data. In this interface, a user assigns the rows to a range of values for one parameter, and the columns to the values for another; each cell then shows the result of applying the parameter values corresponding to its row and column.

However, even with the side-by-side viewing supported by a tabular spreadsheet, it is not clear that this assures good support for multiple scenarios. A layout defined strictly in terms of a two-dimensional table is likely to be inconvenient for representing scenarios defined by variation in more than two parameters, or by arbitrary combinations of parameter settings. A further limitation, that also applies to many non-spreadsheet styles of programming environment, is that any setup of alternatives is relatively static. If a user wishes to switch from examining scenarios that differ with respect to some parameter A to scenarios that differ with respect to parameter B, then C,then perhaps return to A, the repeated redefinition of all the parallel calculations would be decidedly laborious. In discussing the use of what-if scenarios in spreadsheets, Smedley et al. [1996] argued that "current spreadsheets provide little support for this type of interaction" (p. 148). The scenario-management facilities of Microsoft Excel can be seen as an attempt to work around this limitation, by letting the user define a batch of scenarios whose results are calculated and presented in a dynamically reconfigurable "pivot table". Yet such batch-style processing is the antithesis to interactive definition and use of multiple scenarios, so we cannot categorize this as an instance of subjunctive interface.

Another research area related to the aims of subjunctive interfaces is that of extending undo facilities. As pointed out by Terry and Mynatt [2002b], undo is one of the few interface mechanisms that have won general acceptance for supporting open-ended tasks. It facilitates exploration by virtue of its support for revisiting choices or actions that turned out to be unsatisfactory. The simple undo mechanism, however, has a number of limitations [Berlage 1996; Edwards et al. 2000]. One of these is that to undo an earlier choice, you are also undoing the subsequent commands. Another is that the simple undo models do not work well for applications for which users might want to undo at different levels, say in an isolated part of the application. The work on extending undo to support this is similar to the principle in subjunctive interfaces of making available several scenarios that may be manipulated independently or in combination. A few interfaces have been proposed for visualising the different levels or branches in undo graphs, in effect tackling the problem of how to show scenarios side by side [Kurlander and Feiner 1990; Hightower et al. 1998; Derthick and Roth 2000].

In summary, subjunctive interfaces, in contrast to the existing interfaces reviewed above, provide interface capabilities that at the same time (a) are general, in the sense that they can be implemented for a range of applications and data types, (b) allow working with multiple scenarios, and (c) allow interactive control over those scenarios. However, to our knowledge, the usability of subjunctive interfaces has only been evaluated in two studies. Terry et al. [2004] conducted think-aloud sessions with three users of Parallel Paths. The reception of the users was positive, but the study identified problems in their ability to distinguish variations, and it was not intended as, nor provided, a systematic evaluation of usability. The other evaluation was our initial analysis of two studies on the census browser [Lunzer and Hornbæk 2004]; below we summarise those two studies, and provide additional empirical data from a third study so as to investigate more fully the usability of one example of subjunctive interface. We do this because the value of an in-depth, iterative exploration of one example is expected to give us a better understanding of the principles compared to less detailed studies across a range of examples.

3. STUDY #1

All our studies were based on the browsing of census data as described in Section 2.1.1. For Study #1, we had formed the following hypotheses regarding the use of a subjunctive interface rather than the simple one:

- H1: With the subjunctive interface, users would rely less on writing interim marks and notes, for example to keep track of the results of interest, or progress in performing a complex task. This hypothesis reflects the goals of information visualization to amplify cognition through support of external cognition [Card et al. 1999] and of spreadsheet research to improve efficiency [Jones et al. 2003; Chi et al. 1998].
- H2: Users would complete tasks with fewer mouse clicks when using the subjunctive interface [Lunzer and Hornbæk 2003]. As a consequence, we also expected users to be faster with the subjunctive interface.

- H3: Users would have higher satisfaction with and prefer the subjunctive interface. In particular, we expected users to appreciate the direct on-screen comparisons possible with the subjunctive interface.
- H4: Users would report lower mental workload using the subjunctive interface, because they do not have to remember values they wish to compare.

3.1 Interfaces

The simple interface was as shown in the upper part of Figure 1, though the subjunctive interface in Figure 1 is that used for Study #2. For this study, the subjunctive interface had the features described under "Census browser for Study #1" in Figures 4 to 9.

3.2 Subjects

Twenty paid subjects participated in the study: 16 men and 4 women. Subjects were recruited among students and faculty, and had a mean age of 32 years.

3.3 Tasks

The tasks used by Hochheiser and Shneiderman [2000], who previously studied interaction with the census data we are using, were all two-case comparisons that could be answered in about 30 seconds. Although some users might only need to perform simple tasks like these, there may also be tasks requiring longer interaction sequences, iterating over more data. In evaluating the relative strengths of alternative interfaces, it would be risky to generalize from observations of only simple comparisons, so for this study we defined more complex retrieval and comparison tasks, of the following three types:

—*Intra-Set Comparison:* These tasks require pairwise comparisons between many combinations of the records in some small set. For example, one task asks: "Considering Wholesale Trade in [five named counties] in 1993, find how the counties are ordered in terms of number of Employees. In order from fewest Employees to most, what are the Payroll values for these counties?"

—*Iterative Examination:* These tasks call for examination of records that lie in a repeating pattern. For example, "In 1996, for which of the industries do [three named counties] all have 1000 or more Employees?"

—*Iterative Comparison:* These tasks are similar to iterative examination, but call for comparison between the records rather than merely checking whether each record satisfies some criterion. For example, "In which counties does the Payroll for Wholesale Trade fall in every year from 1993 to 1996?"

We expected that, for each task type, appropriate use of the subjunctive interface would provide some benefit over using the simple interface. For intra-set comparisons the benefit is merely in being able to keep values on view rather than having to remember them or write them down. For iterative examinations and comparisons, the iteration can be performed more efficiently if the user first sets up scenarios that express the repeating pattern required by the task.

3.4 Design and Measures

The experiment used a within-subjects design, where each subject first solved a set of nine tasks (three of each of the above types) using one interface, then solved another nine with the other interface. The order in which subjects used the interfaces and the order of the two task sets were systematically varied; each subject was assigned randomly to one of four groups determining these orders.

The independent variables were the two interfaces (simple vs. subjunctive), and the three task types. The dependent measures were the following:

—accuracy in performing tasks, measured as the percentage of tasks for which the correct answer was found;
—number of additional marks and notes made on the answer sheet;
—task completion times, excluding time to read the task;
—subjective satisfaction, measured by five questions from the Questionnaire for User Satisfaction (QUIS) [Chin et al. 1988];
—experienced mental workload, measured by NASA's task load index (TLX) questionnaire [Hart and Staveland 1988];
—preference, measured by asking at the end of the study which interface the subject preferred.

We logged with time stamps all interface actions (mouse clicks and keystrokes) performed by the subjects.

3.5 Procedure

Upon arriving, subjects filled in a questionnaire with background information. Next, they received a standardized oral explanation of how to use the interfaces, and completed four simple training tasks. Any questions or misunderstandings brought up at this stage were resolved. In all, training took an average of 29 minutes.

For each interface, subjects received one task at a time on a piece of paper on which they could also write the answer. Subjects could request clarification of what information the task demanded, but not how to find or record that information. They were given a maximum of eight minutes for each task; if the task was not completed within this time, the subject was asked to move on to the next. Subjects could also decide to abandon a task. Out of the total 360 tasks, two were abandoned and one timed out; these three cases have been excluded from the statistical analysis.

After completing all the tasks for an interface, subjects were given five questions from the QUIS [Chin et al. 1988] and an opportunity to comment on that interface. Next, they completed NASA's TLX questionnaire [Hart and Staveland 1988] as a measure of mental workload. Between using the two interfaces, subjects were given a five-minute break.

After using both of the interfaces, subjects wrote down which interface they preferred and why.

Table I. Usability and Usage Differences Between Interfaces in Study #1

	Simple Interface ($N = 180$)	Subjunctive Interface ($N = 177$)
Percent correct answers	89% (31)	86% (35)
Number of marks written on paper*	2.62 (4.74)	.83 (2.84)
Number of interface actions*	34.1 (21.3)	21.9 (16.0)
Task completion time (s)	135.0 (64.2)	138.4 (72.2)
	($N = 20$)	($N = 20$)
Preference*	2	18

Note: Parentheses give the standard deviation. Asterisks indicate significant differences with $p < .001$.

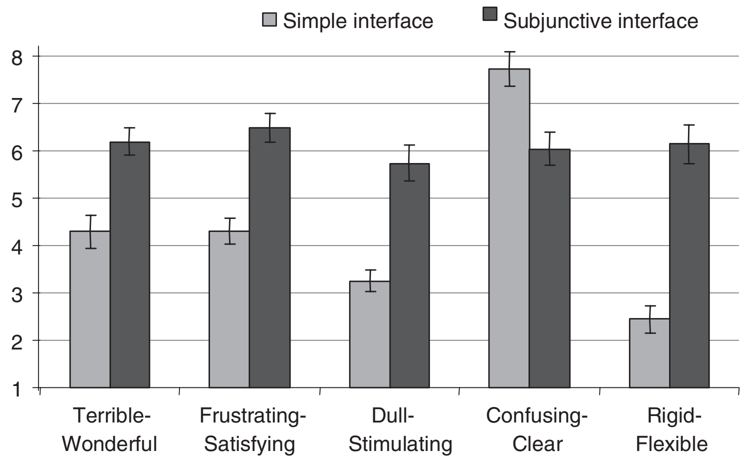


Fig. 10. Subjective satisfaction with the interfaces in Study #1. On all questions there exists a significant difference between interfaces (high scores associated with positive words).

4. RESULTS OF STUDY #1

Table I, Figure 10 and Figure 11 summarize the outcome of the study. Below we analyze each dependent variable in turn, using analysis of variance with repeated measures.

4.1 Accuracy

We find no overall difference in accuracy between the interfaces, $F(1, 19) = 1.14$, $p > .3$. With the subjunctive interface, subjects correctly answered 86% of the tasks; with the simple interface, 89%.

4.2 Number of Marks Written

In the course of answering a task, subjects using the simple interface made more marks on the answer sheet than subjects using the subjunctive interface, $F(1, 19) = 17.32$, $p < .001$. Counting as a single mark anything with an isolated meaning—whether a simple tick or a written data value—subjects using the simple interface made 2.62 marks per task, compared with 0.83 marks for the subjunctive interface. Furthermore, marks were only used in

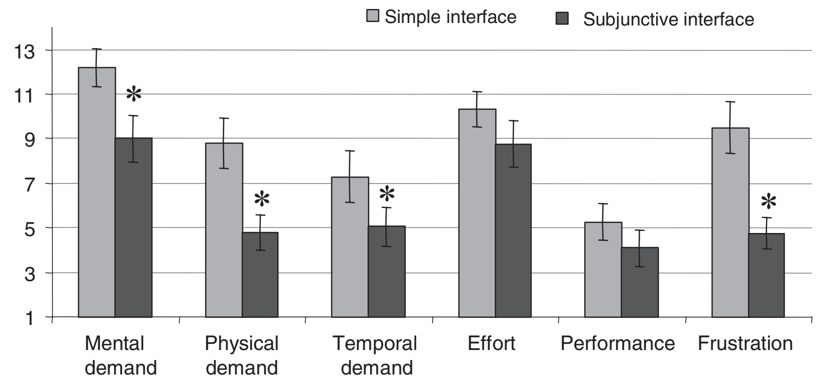


Fig. 11. Subjective workload with the interfaces in Study #1. A significant difference that favors the subjunctive interface was found for the four questions that are marked with asterisks (high scores associated with high workload).

approximately 10% of tasks solved with the subjunctive interface, as against 32% with the simple interface. Note that when explaining the procedure of the study to subjects, we did not mention the option of making marks on the paper.

4.3 Number of Interface Actions

Overall, the data confirmed our hypothesis that the subjunctive interface would require fewer actions for task completion, $F(1, 19) = 81.71$, $p < .001$. On average, subjects with the subjunctive interface used 22 actions to complete a task; with the simple interface, subjects used 34 actions.

4.4 Task Completion Time

Surprisingly, the lower number of interface actions with the subjunctive interface did not lead to significant differences in task completion times, $F(1, 19) = 0.3$, $p > .8$; hypothesis H2 is not confirmed. On analyzing individual tasks, we find differences for three tasks of the most complex type (iterative comparisons). For two of these, completion times were reduced by 47% and 34% when using the subjunctive interface, while conversely the third required 56% more time with the subjunctive interface.

4.5 Subjective Satisfaction

Eighteen subjects preferred the subjunctive interface and two preferred the simple interface—a highly significant and large difference: $\chi^2(1, N = 20) = 12.8$, $p < .001$.

Figure 10 summarizes the subjects' responses to the subjective satisfaction questions. On four questions, the subjunctive interface was rated significantly higher than the simple interface; on one question (how confusing or clear the interface is) the simple interface was rated higher. All these differences are significant using paired t-tests with Bonferroni adjustment (terrible-wonderful: $t = -5.25, p < .01$; frustrating-satisfying, $t = -5.40, p < .01$;

dull-stimulating, $t = -5.42$, $p < .01$; confusing-clear, $t = 3.61$, $p < .05$; rigid-flexible, $t = -7.96$, $p < .01$).

On NASA's TLX, subjects assessed the subjunctive interface as requiring less workload on all dimensions, as shown in Figure 11. Overall there was a significant difference between interfaces, suggesting that the subjunctive interface required less mental workload, $F(1, 19) = 12.14$, $p < .01$. Individual tests show significant differences for the item on mental demand ($t = 2.51$, $p < .01$), physical demand ($t = 3.45$, $p < .01$), temporal demand ($t = 2.27$, $p < .05$), and frustration ($t = 3.15$, $p < .01$).

The comments given by subjects on post-study open-ended questionnaires corroborate the above results. Ten subjects commented that the subjunctive interface supported easy comparison, required less remembering of values, and reduced the need to write values down. One subject wrote: "it was clearly an advantage to be able to see more [values] at once, so that you did not have to do maths in the head or count using your fingers." Similarly, four subjects commented that the main drawback of the simple interface was the need to remember. Another frequent comment, made by seven subjects, was that they had too little time to learn the subjunctive interface. For example, one subject said, "I felt that I need more time to be familiar with the [subjunctive] interface to be able to work faster and have a higher satisfaction."

4.6 Discussion of Study #1

Three of our hypotheses were confirmed. Subjects using the simple interface clearly relied more on pen and paper to remember values and to organize their search, confirming H1. Subjects preferred the subjunctive interface and reported markedly lower mental workload with that interface, confirming H3 and H4, respectively. However, although the average number of interface actions per task was significantly lower for the subjunctive interface, the task-completion time aspect of hypothesis H2 was not supported.

It seems that the overall reason for the subjunctive interface's unimpressive timings is that many subjects encountered difficulties in using it. From notes taken during the study, and later analysis of the detailed interface logs, we defined difficulties as being of two kinds: strategy-formation problems and strategy-execution problems.

4.6.1 *Strategy-Formation Problems.* To solve a task efficiently, the user must first decide on a good strategy for using the available interface. When using the subjunctive interface, many subjects had difficulty figuring out such a strategy. In particular, optimal solution of the iterative tasks requires understanding the practice of setting up scenarios that differ in one parameter, then iterating all of them through the values on a second parameter.

We illustrate these problems on one task that elicited a wide range of successful and unsuccessful responses. The task asked: "Anne Arundel, Carroll, Harford and Howard are four counties in the region called Central Maryland. In 1996, which of these counties had over 10,000 Employees in three or more of the industries?" Figure 12 shows the timing for the ten subjects who tackled this task using the subjunctive interface.

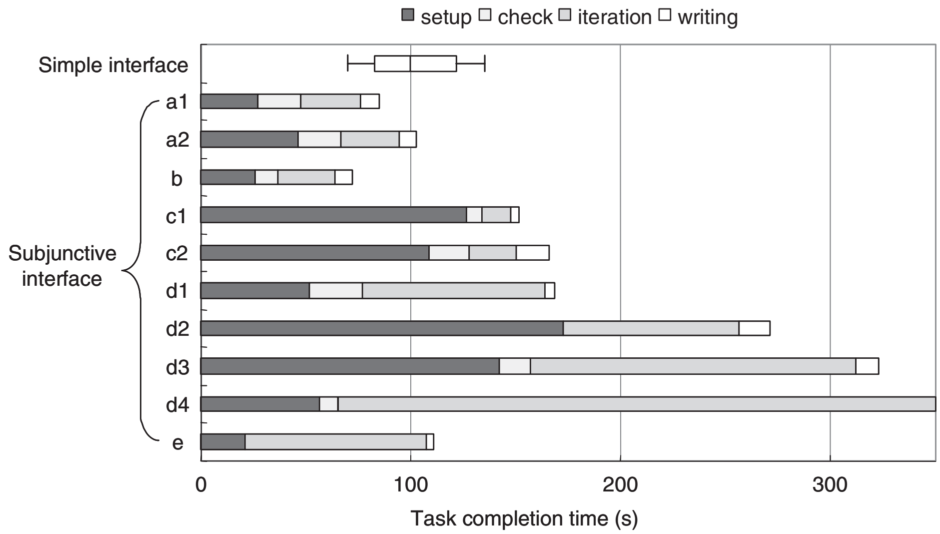


Fig. 12. Timing for ten subjects' solutions of an iterative-examination task in Study #1. Progress in answering the task using the subjunctive interface has been divided into four commonly seen phases: setup to create the scenarios for iteration; check to confirm by inspection that these scenarios reflect the task instructions; iteration when iterating over the required range; and writing between the last interface action and asking for the next task. The box plot at the top shows the times for answering this task with the simple interface.

For this task, an effective strategy is to create a nine-scenario display showing the records for all the industries in one county. The nine Employees values can rapidly be checked by eye to see whether three or more meet the required criterion, and the entire display can be iterated through the other counties. Just two of the ten subjects (a1 and a2 in Figure 12) devised this strategy from the outset and successfully pursued it to find the correct answer.

A less effective strategy is to create scenarios for each of the counties, then iterate that pattern through the industries. The subject must keep a mental or written record of whether three qualifying Employees values have been seen yet for each county. Figure 13 represents the range of county and industry statistics visible using this and the previous strategy. The quality of the strategy does not depend on the number of scenarios set up or mouse clicks needed, but whether the user can see at one time enough data to complete part of the task. Unfortunately, seven of the subjects began their approach to the task by creating the four-county setup. One subject (b) successfully carried the strategy through just using his memory; though cognitively demanding, this was the fastest of all the subjunctive-interface solutions. Two others (c1 and c2) set up the four county scenarios then paused, reset the interface and restarted using the nine-industry approach instead; this is seen in their long setup times but short iterations. Four subjects (d1 to d4) did not make this strategy switch, but apparently tried to adapt the four-county setup to keep on view the records that turned out to meet the task criteria. This was not a effective strategy. Of these four subjects, d4 eventually abandoned the task; the others ended up

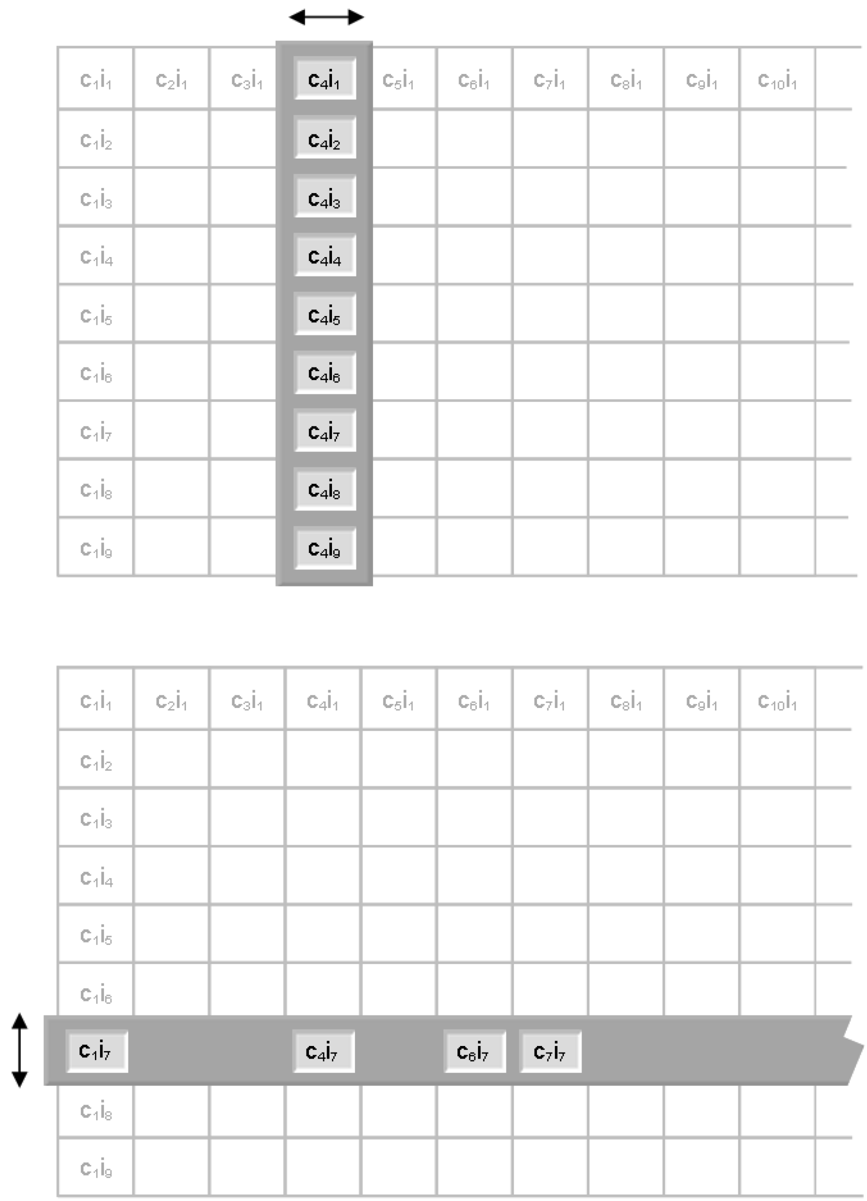


Fig. 13. Schematic comparing strategies for the task being addressed in Figure 12. The grid represents the space of counties (horizontal, truncated) against industries. The upper diagram represents the values that can be seen simultaneously using a setup of nine scenarios for the nine industries, while the lower represents a scenario setup for the four counties referred to in the task. In this task, which requires tallying across the industries for each given county, the former is more efficient.

iterating multiple times, as they might have done with the simple interface. One subject (e) pursued an unusual task-specific strategy, creating scenarios to hold what had turned out to be the largest industries in the first examined county—presumably in the hope that these would also give a quick positive answer for other counties.

4.6.2 *Strategy-Execution Problems.* In many cases, subjects who had apparently decided on an effective strategy then ran into difficulties executing it. In particular, subjects often appeared to be confused by the active-scenarios mechanism (Figure 8(a)), and by the policy of displaying only a single result value if all the scenarios had that value (Figure 6(a)). These and other difficulties, which are discussed in detail in our earlier paper [Lunzer and Hornbæk 2004], guided our redesign of the interface for the next study.

5. STUDY #2

Many subjects in Study #1 commented that the brief scope of the study did not give them time to become proficient with the subjunctive interface. We therefore designed a second study in which subjects would use the interface over multiple sessions, so as to explore whether more experience with the interface would reduce the incidence of strategy-formation and strategy-execution problems that had slowed task completion in Study #1. Our main hypothesis was the following:

- H5: over sessions, subjects using the subjunctive interface would become significantly faster than subjects using the simple interface.

5.1 Interfaces

The simple interface for this study supported the same operations as in Study #1. The subjunctive interface incorporated design changes suggested by our analysis of common strategy-execution problems seen in Study #1. The differences between the subjunctive interfaces are detailed in Figures 4 to 9; the most significant differences in terms of interaction mechanisms relate to the approach to active scenarios (Figure 8), and the influence of this on scenario creation (Figure 4) and scenario update (Figure 9).

5.2 Subjects

Seven subjects who participated in the Study #1 and who were willing to participate again were paid to take part. Study #2 took place about two-and-a-half months after Study #1. With this interval between studies we expected subjects not to have remembered enough detail of the Study #1 tasks to affect their performance.

5.3 Design and Procedure

The study consisted of five sessions, each separated by at least a day. As in Study #1, subjects used both interfaces in each session. In Session 1, subjects were trained and completed nine tasks with each interface. Sessions 2 to 4 were completed at a location of the subjects' choosing. They received a CD with the software, and stapled sets of papers containing nine tasks for each interface; on completing each session, they mailed us the generated log file. In Session 5, subjects came back to the lab and completed the tasks and the subjective-satisfaction questionnaire used in Study #1.

5.4 Tasks

The tasks used for Sessions 1-4 in Study #2 were similar to those used in Study #1. To avoid subjects simply remembering the answers from one session to the next, we constructed four sets of tasks with the same structure as in Study #1 but concerning different counties, industries, years, and kinds of comparison (e.g., finding the year with the highest payroll instead of the lowest employment). Because the tasks were isomorphic we expected only minor differences in completion times between them. Session 5 used exactly the same tasks as Study #1.

Out of the total 630 tasks, three cases have been excluded from the statistical analysis because of subjects timing out or abandoning the tasks.

6. RESULTS OF STUDY #2

6.1 Accuracy and Subjective Satisfaction

Similar to Study #1, subjects answered 95% of the tasks correctly with the subjunctive interface and 93% with the simple interface. We do not find any significant differences between interfaces, $F(1, 6) = 1.21, p > .3$.

The subjects' satisfaction ratings were all in the same direction as in Study #1, favoring the subjunctive interface. On the QUIS filled out after Session 5, three out of five questions were significant using paired t-tests with Bonferroni adjustment (terrible-wonderful: $t = -4.50, p < .01$; dull-stimulating, $t = -7.78, p < .01$; rigid-flexible, $t = -4, 94, p < .01$). In contrast to Study #1, interfaces were not different on the confusing-clear scale ($t = 1.59, p > .1$). This suggests that the longer experience with the interface and our simplification of the facilities for scenario setup made the interface easier to understand and use.

6.2 Task Completion Time

Having corroborated the general results of Study #1, we focussed on task completion time. Figure 14 shows the average task completion time over sessions. Overall, analysis of variance using repeated measures shows that the subjunctive interface was faster than the simple interface, $F(1, 6) = 8.27, p < .05$. Planned comparisons show this to hold for all sessions except Session 1.

In Session 4, subjects were on average 18% faster with the subjunctive interface ($M = 89$s, $SD = 39$) than with the simple interface ($M = 109$s, $SD = 46$), $F(1, 6) = 9.58, p < .05$.

Comparing Session 5 to Study #1, which used the same tasks, we find a large improvement in task completion times for both interfaces. With the simple interface, subjects improved around 19% (Study #1: $M = 135$s, $SD = 64$;

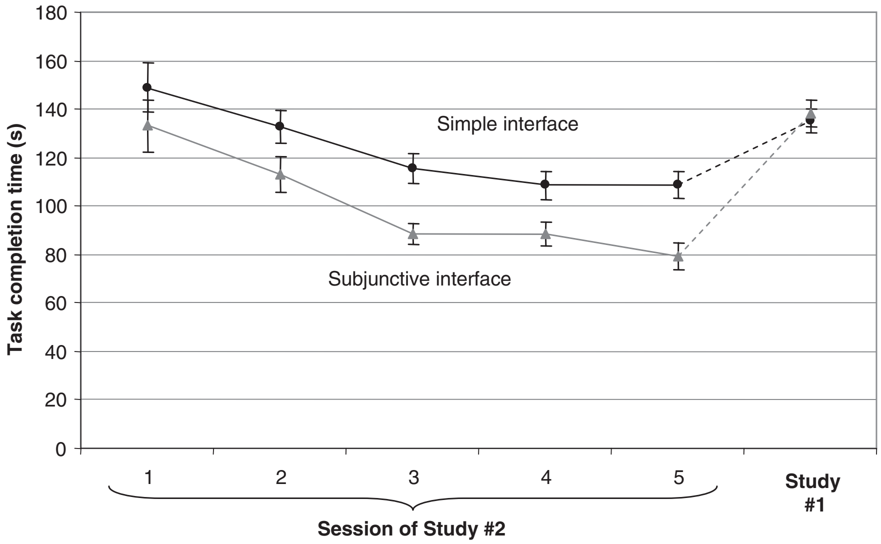


Fig. 14. Task completion times in Study #2 and Study #1. Error bars indicate standard error of the mean. Each session consists of N = 126 observations.

Study #2, Session 5: $M = 109$s, $SD = 45$). With the subjunctive interface, subjects improved 43% (Study #1: $M = 138$s; $SD = 72$; Study #2, Session 5: $M = 79$s, $SD = 44$). While subjects apparently learned to control both interfaces better, this was most pronounced for the subjunctive interface. In Session 5, the subjunctive interface was approximately 27% faster than the simple interface, $F(1, 6) = 208.87$, $p < .001$.

In Study #1, we found differences at the task level favoring either the subjunctive or the simple interface; in Session 5 of Study #2, six tasks were completed significantly faster with the subjunctive interface, and in no tasks were subjects significantly faster with the simple interface. For one task, subjects on average took more than 2.3 times as long with the simple interface.

6.3 Discussion of Study #2

The results of Study #2 strongly confirm our hypothesis H5: over sessions, subjects became faster with the subjunctive compared to the simple interface. In particular, strategy formation problems appeared to diminish as subjects became more experienced at controlling the subjunctive interface.

7. STUDY #3

Study #2 showed that, after some experience with the subjunctive interface, subjects became much faster than with the simple interface. The facilities for working with multiple scenarios in both studies #1 and #2 also improved subjects' satisfaction. These studies, however, reveal little about users' thinking surrounding the creation and manipulation of scenarios; that thinking, including users' decisions, rationales and possible misunderstandings, is likely to form a major input to future work with subjunctive interfaces and other multi-state application interfaces. The aim of the third study was therefore to investigate how users of the subjunctive interface appear to be thinking about scenarios as they set them up and manipulate them.

7.1 Subjects

Seven subjects who participated in Study #1 (and four also in Study #2) and who were willing to participate again were paid to take part. Study #3 took place about six months after Study #2 and eight-and-a-half months after Study #1. Given this length of time, we again assumed that subjects would have forgotten most details of the data set; this study also introduced previously unseen data, as described below.

7.2 Tasks

To evaluate more broadly how scenarios are created and manipulated, we asked subjects to address twelve relatively complex tasks. Half of the tasks were from Study #1; to maximize the difficulty subjects would face, we chose the tasks with the highest task completion times. The other half were open-ended tasks, specifically designed to challenge subjects' proficiency with the interface. These tasks were characterized by their large search space and open definition of correct answers. Examples of such tasks included "The method of aggregating statistics was changed in 1998, affecting some of the industries reported here. Find three examples that illustrate the various effects of this change (e.g., a sudden rise in one or more of the reported amounts, a sudden fall, a single outlying year, etc.)." and "Making judgments based only on the statistics from the start and end of a time period is of course risky; the values that you see may be out of line with the changes in between. Find two cases where only observing the 1993 and 1997 values would give a dramatically false impression of the sequence of changes during that period." While these tasks are probably more complex than realistic tasks users would want to solve with the census browser, they help us explore subjects' thinking in setting up and using scenarios.

7.3 Interface and Data Set

The interface that subjects used was similar to the subjunctive interface used in Study #2. However, to have more data available for the tasks given to subjects, we extended the number of years in the data set from four to nine years, so that the data set comprised 1656 records.

7.4 Design and Procedure

In contrast to the previous studies, there was only one interface for subjects to use. In the first session, subjects were invited to the lab where they were introduced to the interface, using an average of 16 minutes. After the introduction, each subject completed twelve tasks. Any questions or misunderstandings brought up at this stage were resolved. The procedure for receiving and answering tasks was similar to that of Study #1.

The second session was also held in the lab. The subjects were first instructed in how to think aloud, using instructions adapted from those of Ericsson and

Simon [1993, Appendix 1]. Next, participants practiced thinking aloud using two simple tasks with the census browser. Then, the subjects received twelve tasks similar to those of the first session. During practice and data collection, we used a reminder to think aloud, of the form 'keep talking'. In addition, we used acknowledgments ('mm hm'), as suggested by Boren and Ramey [2000]. In cases of unclear comments we used a single-word question, restating a part of the unclear comment with an interrogative intonation, for example as in Boren and Ramey [2000, p. 275]: "Participant: That was odd ... // Practitioner: Odd?" After subjects had completed half the tasks this way, we began probing more actively their considerations surrounding creation and manipulation of scenarios. When subjects were setting up scenarios in a complicated or unclear way, we asked questions to the effect of "Why did you do that?" or "What are you hoping to achieve?" Despite the warnings of Ericsson and Simon [1993] against explanations as a valid source of data, we follow the suggestion of Chi [1997] and consider subjects' comments and reasoning about their interaction with scenarios to be relevant, given our research question. Finally, participants were debriefed. All verbal communication was done in Danish, the native language of the participants. The second session was video taped and all interactions with the interfaces were logged. On average, this session lasted 55 minutes.

7.5 Analysis

To analyze the verbal data we employed techniques similar to those proposed by Chi [1997]. First, we transcribed the videos. Second, we segmented the videos based on the verbal data and on interactions. Third, one of the authors coded activities, thinking, and misunderstandings surrounding the creation and manipulation of scenarios. In addition, the various strategies involved in solving the tasks were coded. Fourth, the other author consolidated the coding, discrepancies were discussed, and a final set of codes to be used was developed.

The outcome of the above analysis was a collection of 236 video clips from the interaction, each reflecting one interesting or problematic aspect. Because of the uncertainty in interpreting the behavior and thinking aloud of subjects, we were conservative in applying codes. The numbers given below, concerning the percentage of tasks in which a certain kind of observation was made, thus represent lower estimates.

8. RESULTS OF STUDY #3

Below, we describe the main findings of Study #3 (see Table II for a summary), illustrated with examples of subjects' think-aloud comments and interactions (Tables III to VI). We focus on findings that are likely to be relevant to application domains and tasks beyond those used in this study.

Section 8.1 describes the overall strategies used by subjects; Sections 8.2 to 8.5 discuss in more detail how subjects set up and manipulated scenarios; Section 8.6 contains a discussion of the findings, including suggestions for how to improve the interface.

Table II. Main Observations of Study #3

Observation	*N*	%
Errors in manipulating scenarios, such as difficulties in hitting labels or in dragging markers	35	42
Choices between complex setups and longer series of interactions	29	35
Rearranging scenarios to fit task, such as using the position of scenarios on multiplexers to reflect the comparisons to be performed	29	35
Sampling values, for example in order to minimize the number of scenarios needed	21	25
Using color as a mnemonic for scenarios	18	21

Note: N refers to the number of task attempts in which the observation was made; the rightmost column gives the percentage with respect to the total number of task attempts (84).

8.1 Strategies

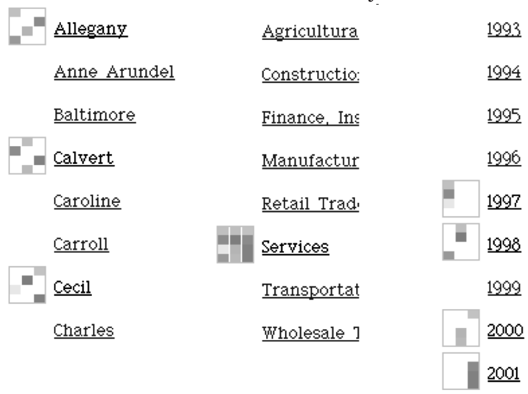
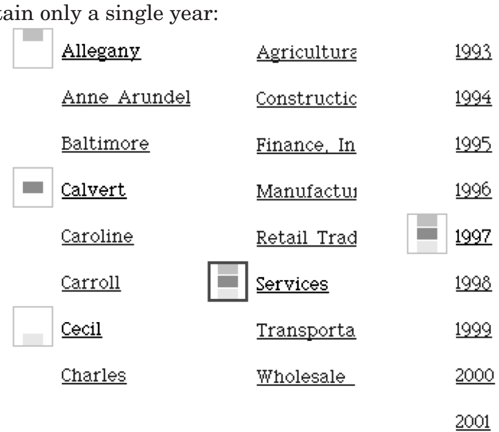
A key issue to solution of most tasks was whether to employ a complex setup of scenarios or a longer series of interface operations; around one-third of the task attempts contain behavior or comments related to this trade-off. In some cases the issue is revealed in subjects' remarks about the complexity of their current scenario setup—remarks such as, "again the question is how much one can get an overview of at once", and "perhaps it would be easier to take just one industry at a time". In other cases, subjects chose to simplify the setup even though this would require them to perform more operations. As one example, we show the interaction in Table III, where it appears that the subject was trying to do too complex a comparison with the subjunctive interface, and thus had to simplify her strategy.

A related issue is whether to intersperse the setting up of scenarios with examination of their contents, or to set up many scenarios then examine them as a batch. This issue is illustrated in subjects' strategies for the following task: "For Retail Trade in 1993, which three counties had the fewest Employees?" Two strategies stand out: (a) setting up just four scenarios, and iteratively moving the scenario with the least interesting (i.e., highest) employees value to the next unexamined county; or (b) setting up scenarios for the first twelve counties, deleting all but the three with the lowest employees values, using the remaining scenarios to examine the next nine counties, and so on. Some subjects seemed to have a strong preference for strategies like (b). One subject remarked "Perhaps I should look at fifteen of these at the same time [fifteen counties with a scenario for every year]. Is that possible?" Had there been no limit, this subject might have created 135 scenarios, and could then have completed the task just by visual scanning with no further interaction. By contrast, some subjects seemed to prefer strategies like (a), perhaps because they involve simpler scenario setups.

Subjects' strategies were affected by only being able to set up twelve scenarios; in 16 task attempts (19%) this limitation was mentioned by subjects. Some subjects seemed to perceive the twelve-scenario capability as a resource to be used to the full, scanning as broad a sweep as possible (e.g., handling three industries over four years). Conversely, as seen in Table III and noted above, in some cases twelve scenarios was too many to work with.

Compared to Study #2, we deliberately assigned hard tasks; many of them would be prohibitively arduous to address with the simple interface. An

Table III. Simplifying the Setup of Scenarios

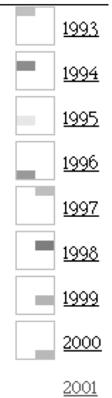
<table>
<tr><th>Subject's Thinking Aloud</th><th>Subject Interactions and Our Interpretation</th></tr>
<tr><td>There are quite big differences here, between two years</td><td>The subject is addressing a task that asks 'Find a group of three counties for which, in some industry, the Payroll ranking of the counties changes several times over the years 1993 to 2001'. She has set up 12 scenarios, for all combinations of three counties and four years:

</td></tr>
<tr><td>Perhaps one should ... no, I have to look at at least three counties to be able to compare</td><td></td></tr>
<tr><td>Perhaps it is easier if I take slightly fewer years</td><td></td></tr>
<tr><td>I actually think it is a little hard to overview</td><td>After this remark, the subject goes on to simplify the setup to contain only a single year:

</td></tr>
</table>

indication that, even with the subjunctive interface, these tasks were hard is that, in 19 task attempts, subjects chose to make notes on the paper, writing down values, colors of scenarios, and counts.

8.2 Reasons for Setting Up and Keeping Scenarios

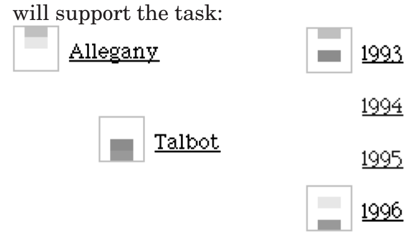
We had expected scenarios to be set up mainly for comparisons and for efficient parallel update. While such use took place in almost all task attempts, from the videos we identified various additional features of scenario use.

Table IV. Choosing a Sampling of Values, Rather Than the Full Range

Subject's Thinking Aloud	Subject Interactions and Our Interpretation
	Starts out by creating eight scenarios, each corresponding to a year 
Actually I think I will take	
How many have we got?	Looks at the number of years
I think I will select a more representative sample of the years	Deletes four years, ending up with four scenarios 
I can always add more later	After having found three candidate counties for the answer to the task, the subject moves the markers in the year menu, checking the years that were not present in the four scenarios selected initially

In 25% of the task attempts, subjects limited the number of scenarios by using a sampled subset of the values in a menu. One kind of sampling involved setting up a few (often three or four) markers within, say, the year menu, then adjusting the other menus to find scenarios that appeared to have the desired year-dependent properties, and finally moving the year-menu markers to confirm those properties in the other years; Table IV shows an example of this. Another kind of sampling involved dynamically moving a marker over a range of menu items while watching the result displays, for example to gather an overall impression of counties' sizes before picking a few to explore in detail.

Table V. Use of Color as a Mnemonic for Scenarios and Their Corresponding Set of Parameters

Subject's Thinking Aloud	Subject Interactions and Our Interpretation
We now need to have one of them higher in 96 compared to 93 and the other the other way around	A restatement of the task description. 'one of them' refers to a county.
So now I just have to find industries where the middle numbers here are larger than the other numbers	First considers how this setup of scenarios will support the task: 
That is, yellow is larger than grey and blue is larger than red, or the other way around	Then appears to be using the colors as a mnemonic for the comparisons to be done

Less frequently, we observed scenarios being used as storage, that is, used not to compare against but rather to hold some intermediate result. In three task attempts, subjects copied the marker they were using for exploring when they found one of a number of acceptable answers. By doing so, they stored the answer in the original scenarios, continuing their exploration using the new scenario copies.

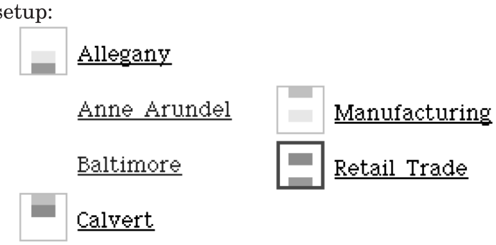
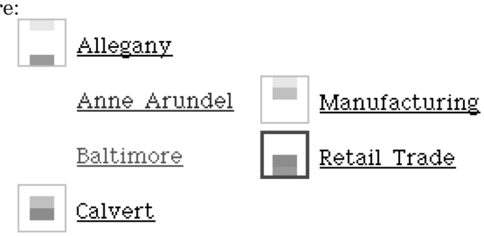
Occasionally, scenarios were set up and kept in the interface but not used for anything. In one of three cases, a subject searching for counties in which some industry could be seen to grow with each successive year set up scenarios for one county, a sample of four years, and an industry. She then copied that county's marker (and its four scenarios) to two other counties, for a total of twelve scenarios. For the remainder of the task, however, the subject only moved one of the county markers. Thus, eight of the scenarios simply cluttered the display, presumably making comparisons more difficult.

8.3 Rearranging Scenarios

Subjects often used the interface's facilities for rearranging scenarios (by dragging the views within a result display) to create a layout that helped to reveal the information requested by a task. Some kind of rearrangement of scenarios took place in about one-third of all task attempts.

As one example of this behavior, Table VI shows one subject's remarks on starting a task that asked "Can you find any counties that in 1996 had fewer Establishments in Manufacturing but more in Retail Trade than did Calvert?" This subject went to some effort to lay out the scenarios so that he would just need to check if values were increasing from multiplexer position 1 (the top position) to 2, and from position 3 to 4. The feeling of having transferred some of the task's complexity to the layout was expressed in the remark "And then all I need to do is move around here [i.e., move a menu marker]". Approximately 14% of the task attempts contained some rearranging of scenarios with a similar purpose.

Table VI. Use of Multiplexer Layout to Make Tasks Easier

<table>
<tr><th>Subject's Thinking Aloud</th><th>Subject Interactions and Our Interpretation</th></tr>
<tr><td></td><td>Starts by setting up two scenarios, one for Calvert + 1996 + Manufacturing and one for Calvert + 1996 + Retail Trade</td></tr>
<tr><td>And so we are just creating one of these</td><td>Copies the two scenarios to another county, obtaining this setup:

</td></tr>
<tr><td>And it came up here</td><td>Referring to the fact that the two old scenarios are now at the top of the multiplexer and the two copies are at the bottom</td></tr>
<tr><td>Then it would probably be advantageous to think a bit</td><td></td></tr>
<tr><td>Fewer in manufacturing, that is to say, in the yellow</td><td>First refers to the description of the task and next to a scenario that contains manufacturing and is color-coded yellow (at position 3 in the above setup). He then moves that scenario to the top of the multiplexer</td></tr>
<tr><td>Then I sort them so that it should be increasing here</td><td>Moves the scenarios so that the increase should be from position 1 to position 2 of the multiplexer, as shown here:

</td></tr>
<tr><td>And hmm it should be increasing here</td><td>Similarly confirms that the values should be increasing from position 3 to position 4 of the multiplexer</td></tr>
<tr><td>And then all I need to do is move around here</td><td>The subject appears to consider the rest of the task easy, because part of the challenge has been mapped onto the setup of scenarios</td></tr>
<tr><td></td><td>Subject finishes the task by moving a marker through all counties, comparing them to the county Calvert.</td></tr>
</table>

We also saw five task attempts in which subjects used rearrangement not based on the scenarios' input values, but according to the values in one of the result displays. As an example, when ordering the scenarios according to their Employees values one subject remarked "I want to rank [the scenarios] and see if [the ranking] is similar for the next [county]". Switching all the scenarios to the next county, it would immediately become clear whether the same industry ranking applies in this county.

In tasks that called for multiple pairwise comparisons between values, subjects in three cases found that the scenario layout did not present the comparison pairs in useful juxtaposition. In such cases the subjects rearranged the layout so that the pairs were horizontally or vertically adjacent, thus making the comparisons easier. The census browser's multiplexers have a column-oriented layout, with each column holding four views, and subjects discovered that it is easiest to create tidy alignments of scenarios when there is a multiple of four.

8.4 Referring to Scenarios

In 21% of the task attempts, while interacting with the interface subjects referred to the scenarios in terms of their colors on the result multiplexers and on the markers. We believe that this was not just because a color name is a convenient shorthand when thinking aloud, but because users were thinking in terms of scenarios and hence colors during the course of a task. For example, one subject noted "No, this doesn't qualify; it is meant to be higher—blue should be higher", where the blue scenario had been set up to show the 1996 statistics for a particular county. When iterating through a range of values, such as the industries, some subjects would repeat in terms of scenario colors what relationship between result values they were seeking; Table V shows an example.

Using scenario colors as a mnemonic is useful because subjects do not have to remember the values that comprise a particular scenario, just its color; the benefit seems similar to the use of position to off-load task complexity onto the display.

8.5 Interacting with Multiplexers

Rather than making comparisons by setting up scenarios side by side, subjects would sometimes switch a menu marker quickly back and forth between two values. We observed this behavior in eight task attempts (10%). Even though such comparisons are susceptible to the problems of the simple interface discussed earlier, they are very efficient: each click can switch the contents of up to twelve scenarios. Note that in four of these cases subjects were working with six scenarios or fewer, meaning that they could still—within the twelve-scenario limit—have created enough scenarios to perform the comparison side by side.

We observed nine cases of subjects having difficulties in using the scenarios' color-coding and position to relate values on result displays to their corresponding menu markers, and hence parameter settings. As mentioned above, subjects referred to scenarios extensively by their colors, but in some cases this seemed insufficient to support seamless integration of the menus with the result displays. For example, a subject might pick up a marker believing it to correspond to the lowest value in a given display, but then discover that the wrong scenario had been updated. Some subjects explicitly commented on this challenge, for example, saying "Now I just have to overview which colors correspond to which [scenarios]". Not surprisingly, this problem was worse when many scenarios had been set up. One of the subjects participating in the study was color-blind, but succeeded in using just the position of scenarios to relate the results to the menus.

We analyzed in some detail how subjects manipulated scenarios, and thereby identified a variety of manipulation errors. One or more errors were present in

42% of the task attempts. We recorded fourteen task attempts where subjects had difficulty in hitting the labels; twelve showing difficulty in dragging scenario markers; and three where subjects accidentally activated the pop-up. While all these problems sound serious, they delay the subject only briefly—usually a second or less. The only substantial manipulation problem we identified on the videos happened when subjects mistakenly moved a marker when they intended to copy it; this happened eight times, and delayed subjects somewhat in completing the tasks.

8.6 Discussion of Study #3

The main finding of Study #3 is that the subjects seemed to treat scenarios as information holders, each capable of containing one combination of parameter settings. Subjects often used scenarios' positions or colors as abstractions that persisted through a task, independent of the scenarios' contents. In this way, they transformed tasks involving multi-parameter criteria into straightforward comparisons between values in multiplexers. With few exceptions, the subjects successfully used the subjunctive-interface facilities to complete even the most complex of the tasks within a constrained time, often in creative and effective ways. Indeed, scenarios were used in a richer variety of ways than we had expected, such as for intermediate storage, as samples over the range of a parameter, or for organizing information spatially to reflect the needs of a task. However, subjects did not always choose the most efficient strategy in terms of interface operations: occasionally they chose to proceed by iteration rather than by setting up more scenarios. This may partly reflect their inexperience with the interface, but also points to a trade-off between the cognitive complexity of multi-scenario setups and the effort involved in an iterative approach.

Further benefits obtained from the subjunctive interface were evident, though not explicitly mentioned in subjects' thinking aloud. One was the making of rapid perceptual judgements based on a multi-scenario display—such as noticing that all the results were approximately equal, and thus all met some criterion, or conversely that just one result was an outlier. Another was a willingness to explore beyond the minimum needed to answer a task—such as continuing even after finding a potential result, or double-checking previously seen cases.

However, in Study #3, we still observed instances of what we described earlier (Section 4.6.1) as strategy-formation problems, where some subjects' usage strategies for a given task were less effective than others in reducing the cognitive load. We believe that various factors were involved in subjects' sometimes choosing strategies that turned out to be inefficient. First, because of the nature of the data, for some tasks it was not necessarily clear from the task description what a good strategy would be. Having embarked on one strategy, subjects may have been reluctant to switch to another even if it appeared to be more efficient. Second, the existence of some time pressure may have made subjects want to get started on a task without weighing up alternative strategies, and the desire not to waste time may again have discouraged them from switching. Third, there are apparently individual differences between subjects, with some preferring to set up views containing as much information as possible, while others prefer to use iteration. Fourth, some subjects may have felt that setting up several scenarios, and perhaps later having to remove them, involved more effort than having to remember a few values.

While subjects experienced many manipulation problems, some of which suggest areas for improving the studied interface, we did not see the kind of strategy-execution problems evident in Study #1. We hypothesize that these problems have been designed away in the later versions of the interface.

Based on recurring patterns of interaction, and a number of comments from subjects, we believe that the following may be useful enhancements to the interface: (1) assistance with scenario setup, such as a lightweight mechanism for generating a scenario for each item in a menu; (2) ability to undo any operation, including scenario creation and removal; (3) ability to reorder scenarios by sorting on the values in result displays; (4) ability to iterate through a menu without having to keep one's eyes on the menu itself (e.g., by using arrow or tab keys); (5) ability to move all of a menu's markers at the same time; (6) alternative scenario layout, for example directly supporting two-dimensional (two-parameter) variation.

9. CONCLUSION

Currently, most applications limit users to examining just one scenario at a time, in what Terry and Mynatt [2002a] call the single-state document model. Choices of navigation paths, design alternatives, and parameters to simulations must be specified unequivocally, and their outcomes examined one by one. Consequently, users may be discouraged from attempting to gain a full understanding of the relative merits of their choices, may face cognitive challenges in relating choices to outcomes, and may experience difficulty and boredom in manipulating the application to find an outcome that is satisfactory. Conversely, subjunctive interfaces support (1) the setup of multiple, perhaps mutually incompatible sets of choices as independent scenarios; (2) side-by-side viewing of scenarios, to facilitate comparison between them; and (3) parallel adjustment of scenarios, to facilitate rapid exploration over a wide range of choices.

We have shown how these principles can be implemented in simple applications for information access, real-time simulation, and document design. Although other solutions to the problems addressed by subjunctive interfaces have been developed within specific branches of HCI, those solutions are not generally applicable, for example because they only work on a certain kind of data. We argue that the subjunctive-interface principles represent a general solution that can be applied as an extension to a wide range of existing applications. Furthermore, having demonstrated cases where the extension of the interface consists merely in replacing the standard input and result widgets with versions capable of handling multiple scenarios, we argue that users migrating from the original to the subjunctive-interface version of such an application would probably need only minimal relearning. Standardizing the interface mechanisms for multiple-scenario interaction would bring us closer to the notion, mentioned in the Introduction, that users should expect subjunctive-interface facilities just as they expect to be able to use undo. If this standardization can be embodied in the widgets and callbacks offered by a software toolkit, it may also be the case that equipping an application with a subjunctive interface does not incur substantial extra development costs. However, for the time being, it is too early to define the features that should appear in such a toolkit.

To investigate the usability of the subjunctive-interface approach, we conducted three studies of a census browser extended with a subjunctive interface. The studies show that subjects preferred the subjunctive interface, and rated it as being more satisfying to use. With a simple, baseline interface, subjects also depended to a larger extent on writing down or remembering data, as suggested by more interim marks made on paper and by the higher mental workload reported. In Study #1, we found that with the subjunctive interface tasks were completed with fewer interface actions but no reduction in task completion time. Study #2 showed how, after more practice with the subjunctive interface, subjects were completing tasks 27% more quickly than with the simple interface. In Study #3, we found that subjects were able to generate and pursue strategies that put the subjunctive interface to effective use on more complex, open-ended tasks. Subjects were seen to treat scenarios as information holders with readily changeable contents, and thus transformed tasks with multi-parameter criteria into straightforward comparisons between values within multiplexers. Subjects also used scenarios in a richer variety of ways than we had expected, including for intermediate storage, as samples over the range of a parameter, or for organizing information spatially to reflect the needs of a task. Overall, our results corroborate previous arguments about the usability of subjunctive interfaces [Lunzer and Hornbæk 2003].

We readily acknowledge that there are limits to the generalizability of our results to other kinds of task and other kinds of application. Indeed, even some of the tasks within Study #3 highlight limitations of our existing implementations. Further empirical work is needed to investigate how well the subjunctive-interface approach works for applications with larger numbers of variables and larger data sets, and for complex, perhaps creative tasks, lasting hours or days.

In addition to extending our empirical studies, we see two lines of future work. The first is to reevaluate and generalize the principles of subjunctive interfaces. The subjunctive interface approach as presented here, in common with the work of Terry et al. [2004], addresses only the creation of scenarios as explicitly requested by the user. Another approach is for the interface to be proactive in offering scenarios that are likely to be of value—for example, based on a sampling of choice values [Marks et al. 1997], or based on suggestions from a recommender. The aim of such mixed-initiative setup would be to combine the control and transparency of direct-manipulation interfaces with the potential utility of heuristically generated choices. The subjunctive interface approach also requires that all scenarios be visible at the same time, which creates a challenge in relation to how many scenarios can exist simultaneously. We are reluctant to relax this requirement, which we feel serves an important role in encouraging users to consider the context surrounding their explorations rather than just a narrow focus. However, its impact could be reduced by use of a focus+context visualization that lets the user control the emphasis given to individual scenarios.

A second line of work is to investigate in more detail the psychological effects of using subjunctive interfaces to explore alternatives. Relevant to this are the findings of Terry et al. [2004] regarding how experts work with multiple variations, including the observation that users may decide to explore alternatives before, during or after interacting with some choice setting. Another relevant question is whether well-described phenomena in the psychology of choice, such as anchoring [Tversky and Kahneman 1974], will persist or change with subjunctive-interface support, and how users benefit from the qualitative difference of having interfaces in which specifying some combination of choices does not require the previous choices to be discarded. Interacting with subjunctive interfaces could also be considered in relation to sensemaking [Russell et al. 1993], for example by evaluating whether such interaction can provide benefits in analyzing multivariate data sets. Finally, subjunctive interfaces might gain from being systematically rethought in terms of psychological theories such as those of Peter Naur and William James (e.g., James [1890], Naur [1995], and Frøkjær and Hornbæk [2002]).

Overall, our studies provide the first data supporting the claim that subjunctive interfaces offer a usable and generalizable approach to supporting exploration of alternative scenarios.

ACKNOWLEDGMENTS

We thank Erik Frøkjær for ongoing comments, and Yuzuru Tanaka at the Meme Media Laboratory for supporting our continued collaboration. We also thank Harry Hochheiser for lending us the census data. All the interfaces presented in this article were built using the Morphic library within Squeak Smalltalk.

REFERENCES

Ahlberg, C. and Shneiderman, B. 1994. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. ACM, New York, 313–317.

Berlage, T. 1996. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Trans. Comput.-Human Interact. 1*, 3, 269–294.

Boren, M. T. and Ramey, J. 2000. Thinking aloud: Reconciling theory and practice. *IEEE Trans. Prof. Commun. 43*, 3, 261–277.

Burnett, M., Atwood, J., Djang, R., Gottfried, H., Reichwein, J., and Yang, S. 2001. Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. *J. Funct. Program. 11*, 2, 155–206.

Card, S. K., Mackinlay, J. D., and Shneiderman, B. 1999. *Readings in Information Visualization*. Morgan Kaufmann, San Francisco, CA.

Chi, M. T. H. 1997. Quantifying qualitative analyses of verbal data: A practical guide. *J. Learn. Sci. 6*, 3, 271–315.

Chi, E. H., Riedl, J., Barry, P., and Konstan, J. 1998. Principles for information visualization spreadsheets. *IEEE Comput. Graph. Appl.* (Jul./Aug.), 30–38.

Chin, J. P., Diehl, V. A., and Norman, K. L. 1988. Development of an instrument for measuring user satisfaction of the human-computer interface. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. ACM, New York, 213–218.

DERTHICK, M. AND ROTH, S. F. 2000. Data exploration across temporal contexts. In *Proceedings of Intelligent User Interfaces*. ACM, New York, 60–67.

EDWARDS, W. K., IGARASHI, T., LAMARCA, A., AND MYNATT, E. D. 2000. A temporal model for multi-level undo and redo. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, 31–40.

ERICSSON, K. A. AND SIMON, H. 1993. *Protocol Analysis: Verbal Reports as Data*, revised edition. MIT Press, Cambridge, MA.

FRØKJÆR, E. AND HORNBÆK, K. 2002. Studying the utility of metaphors of human thinking in HCI. In *Proceedings of the 2nd Danish Human-Computer Interaction Research Symposium*. DIKU Tech. rep. no. 2002/19, 19–21.

FUJIMA, J., LUNZER, A., HORNBÆK, K., AND TANAKA, Y. 2004. Clip, connect, clone: Combining application elements to build custom interfaces for information access. In *Proceedings of ACM Symposium on User Interface Software and Technology*. ACM, New York, 175–184.

HART, S. G. AND STAVELAND, L. E. 1988. Development of NASA-TLX: Results of empirical and theoretical research. In *Human Mental Workload* , P. A. Hancock and M. Meshkati, Eds. Elsevier, Amsterdam, The Netherlands, 139–183.

HIGHTOWER, R. R., RING, L. T., HELFMAN, J. I., BEDERSON, B. B., AND HOLLAN, J. D. 1998. Graphical multiscale web histories: A study of PadPrints. In *Proceedings of ACM Conference on Hypertext*. ACM, New York, 58–65.

HOCHHEISER, H. AND SHNEIDERMAN, B. 2000. Performance benefits of simultaneous over sequential menus as task complexity increases. *Int. J. Human-Comput. Interact. 12*, 2, 173–192.

HOFSTADTER, D. R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books, New York.

HUDSON, S. 1994. User interface specification using an enhanced spreadsheet model. *ACM Trans. Graph. 13*, 3, 209–239.

IGARASHI, T., MATSUOKA, S., KAWACHIYA, S., AND TANAKA, H. 1997. Interactive beautification: A technique for rapid geometric design. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST'97)* (Banff, Alb., Canada). ACM, New York, 105–114.

IGARASHI, T. AND HUGHES, J. F. 2001. A suggestive interface for 3D drawing. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (*UIST'01*) (Orlando, FL). ACM, New York, 173–181.

JAMES, W. 1890. *The Principles of Psychology*. Henry Holt, New York.

JANKUN-KELLY, T. J. AND MA, K.-L. 2001. Visualization exploration and encapsulation via a spreadsheet-like interface. *IEEE Trans. Visual. Comput. Graph. 7*, 3, 275–287.

JANTKE, K. P., LUNZER, A., AND FUJIMA, J. 2005. Subjunctive interfaces in exploratory e-learning. In *Professional Knowledge Management: Third Biennial Conference* (*WM 2005*), (Kaiserslautern, Germany) Revised Selected Papers (Lecture Notes in Computer Science, Vol. 3782), K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick and T. Roth-Berghofer, Eds. Lecture Notes in Computer Science, Vol. 3782, Springer-Verlag, Berlin, Heidelberg, New York, 176–188.

JONES, S., BLACKWELL, A., AND BURNETT, M. 2003. A user-centred approach to functions in Excel. In *Proceedings of the 8th ACM SIGPLAN International Conference on Functional Programming* (*ICFP 2003*), ACM, New York, 165–176.

KURLANDER, D. AND FEINER, S. 1990. A visual language for browsing, undoing, and redoing graphical interface commands. In *Visual Languages and Visual Programming*, S. K. Chang, Ed., Plenum Press, New York, 257–275.

LEVOY, M. 1994. Spreadsheets for images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, 139–146.

LUDÄSCHER, I., ALTINTAS, C., BERKLEY, D., HIGGINS, E., JAEGER-FRANK, M., JONES, E. LEE, TAO, J., AND ZHAO, Y. 2006. Scientific workflow management and the Kepler system. *Concurr. Computat. Pract. Exper. 18*, 10, 1039–1065.

LUNZER, A. 1999. Choice and comparison where the user wants them: Subjunctive interfaces for computer-supported exploration. In *Proceedings of IFIP TC.13 International Conference on Human-Computer Interaction*, IOS Press, Amsterdam, The Netherlands, 474–482.

LUNZER, A. 2004. Benefits of subjunctive interface support for exploratory access to online resources. In *Proceedings of the Dagstuhl Workshop on Intuitive Human Interface for Organizing and Accessing Intellectual Assets*. G. Grieser, and Y. Tanaka, Eds., Lecture Notes in Computer Science, Vol. 3359, Springer-Verlag, Berlin, Heidelberg, New York, 14–32.

LUNZER, A. AND HORNBÆK, K. 2003. Side-by-side display and control of multiple scenarios: Subjunctive interfaces for exploring multi-attribute data. In *Proceedings of OzCHI*, IEEE Computer Society Press, Los Alamitos, CA, 202–210.

LUNZER, A. AND HORNBÆK, K. 2004. Usability studies on a visualization for parallel display and control of alternative scenarios. In *Proceedings of Working Conference on Advanced Visual Interfaces*. M. F. Costabile, Ed., ACM, New York, 125–132.

MARKS, J., ANDALMAN, B., BEARDSLEY, P. E., FREEMAN, W., GIBSON, S., HODGINS, J. et al. 1997. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of ACM Conference on Computer Graphics and Interactive Techniques*. ACM, New York, 389–400.

NARDI, B. 1993. *A Small Matter of Programming.* MIT Press, Cambridge, MA.

NAUR, P. 1995. *Knowing and the Mystique of Logic and Rules*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

OINN, T., GREENWOOD, M., ADDIS, M., ALPDEMIR, N., FERRIS, J., GLOVER, J., GOBLE, C., GODERIS, A., HULL, D., MARVIN, D., LI, P., LORD, P., POCOCK, M., SENGER, M., STEVENS, R., WIPAT, A., AND WROE, C. 2006. Taverna: Lessons in creating a workflow environment for the life sciences. *Concurr. Comput. Pract. Exper. 18*, 10, 1067–1100.

RAO, R. AND CARD, S. 1994. The table lens: Merging graphical and symbolic representations in an interaction focus + context visualization for tabular information. In *Proceedings of ACM Conference on Human Factors in Computing*. ACM, New York, 318–322, 481–482.

ROBERTS, J. C. 2000. Multiple-view and multiform visualization. In *Proceedings of Visual Data Exploration and Analysis VII, Proceedings of SPIE 3960*. 176–185.

RUSSELL, D., STEFIK, M., PIROLLI, P., AND CARD, S. 1993. The cost structure of sensemaking. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. ACM, New York, 269–276.

SMEDLEY, T. J., COX, P. T., AND BYRNE, S. L. 1996. Expanding the utility of spreadsheets through the integration of visual programming and user interface objects. In *Proceedings of the Workshop on Advanced Visual Interfaces*. ACM, New York, 148–155.

STOLTE, C., TANG, D., AND HANRAHAN, P. 2002. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *IEEE Trans. Visual. Comput. Graph. 8*, 1, 52–65.

TERRY, M. AND MYNATT, E. D. 2002a. Recognizing creative needs in user interface design. In *Proceedings of 4th Conference on Creativity & Cognition*. ACM, New York, 38–44.

TERRY, M. AND MYNATT, E. D. 2002b. Side views: Persistent, on-demand previews for open-ended tasks. In *Proceedings of ACM Conference on User Interface Software and Technology*. ACM, New York, 71–80.

TERRY, M., MYNATT, E. D., NAKAKOJI, K., AND YAMAMOTO, Y. 2004. Variation in element and action: Supporting simultaneous development of alternative solutions. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. ACM, New York, 711–718.

TOOMIN, M. 2006. Abstraction for end-users. In *The Next Step: From End-User Programming to End-User Software Engineering: Proceedings of WEUSE II Workshop.* M. Burnett, B. Myers, M. Rosson, and S. Wiedenbeck, Eds. http://eusesconsortium.org/weuse/weuseII-proceedings.pdf.

TSANG, S., BALAKRISHNAN, R., SINGH, K., AND RANJAN, A. 2004. A suggestive interface for image guided 3D sketching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (*CHI'04*) (Vienna, Austria). ACM, New York, 591–598.

TUFTE, E. 1997. *Visual Explanations*. Graphics Press, Cheshire, CT.

TVERSKY, A. AND KAHNEMAN, D. 1974. Judgement under uncertainty: Heuristics and biases. *Science 185*, 1124–1130.

TWEEDIE, L., SPENCE, B., DAWKES, H., AND SU, H. 1996. Externalising abstract mathematical models. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. ACM, New York, 406–412.

Received January 2005; revised January 2005 and September 2006; accepted February 2007 by Paul Dourish