

Interfaces that Reduce the Cost of Examining Alternatives

Position paper for CHI 2004 Workshop on Designing for Reflective Practitioners.

Aran Lunzer
University of Copenhagen, Denmark
aran@bigfoot.com

Introduction

My contribution to supporting reflection in the use of computer tools lies in designing cognitively low-cost mechanisms for requesting and working with many scenarios simultaneously. Rather than being stuck with an interface that allows manipulation of just one case at a time, a user can bring many cases together, side by side, and is thus helped in reflecting on the relationships between them.

The work reported here began as my PhD project in the early 1990s. The motivation was the observation that many computer applications (including search engines, document formatters, design tools), though offering great flexibility in the range of results they could deliver, made it hard for users to request and compare different results. Consequently, users of such tools were motivated to satisfice, i.e., to accept the first result that appeared acceptable, rather than to explore further to see whether alternative scenarios may produce results that were better. I felt that if the cost of making such explorations could be reduced sufficiently, this would tip the balance in favour of exploration, and hence lead to better-informed decisions.

This paper summarises the developments and research directions arising from that original motivation.

Result-space reconnaissance

The focus of my PhD (Lunzer, 1996) was the idea of 'result-space reconnaissance' – giving the user a cheap way to examine a wide range of scenarios, by dispatching the computer to evaluate them and to report just summary values that help with evaluation and comparison. The key features of a result-space reconnaissance setup are therefore as follows:

- A user specifies the scenarios that are to be examined – in terms of alternative settings for various input parameters – and the measurements to be taken from each scenario.
- The computer evaluates all specified scenarios, makes the requested measurements on their results, and provides an interactive visualisation that correlates the parameter settings with the measurement values.
- The user works with the visualisation to find scenarios with interesting combinations of properties, and to request the detailed results for those scenarios.

The chosen domain of study was the formatting of documents in \LaTeX , given freedom over such parameters as base font size, line spacing, the scaling of embedded figures, and so on. For what could be argued was a small additional investment in effort, users could have their documents formatted in dozens of alternative ways, and could discover which results had desirable properties such as fitting onto a certain number of sheets, or having the page breaks occur at what the user felt to be convenient places in the text.

The outcome was mixed. Although some test subjects felt that this was a worthwhile trade-off of effort against the potential to uncover excellent formatting results, others were less motivated. Some felt that any output of \LaTeX was probably good enough; to borrow the words of Gerhard Fischer (2002), they simply wanted to be consumers. Furthermore, the extra effort was spent on using the mechanisms for requesting reconnaissance (pseudo- \LaTeX markup), and the interactive parallel-coordinates visualisation of the results – both of which are distractions from a user's core activity of preparing a document. This signalled the need for a more integrated approach to handling alternatives, embedded within the main tool for the supported activity.

Subjunctive interfaces

The 'subjunctive interface' approach (Lunzer 1998, 1999; Lunzer & Hornbæk 2003) was inspired by Hofstadter's (1979) idea of a magical television that could show a viewer not just the broadcasts from a single reality, but from alternative realities that differed in ways freely chosen by the viewer – such as realities involving different weather conditions, or taking place on entirely different planets. The key features of a subjunctive interface are as follows:

- The user of an application should be able to set up multiple scenarios that differ in arbitrary ways. For example, when offered some choice in the application's interface, the user should be able to say 'maybe I want value X , but maybe Y or Z ... so let me set up all three and see how things turn out in each case'.
- The scenarios should be viewable simultaneously, side by side, in a way that helps the user to compare them and also to keep track of all the values – all the inputs, and their corresponding results – belonging to each scenario. The results should be shown using essentially the same display techniques normally used for the single-scenario version of the application.

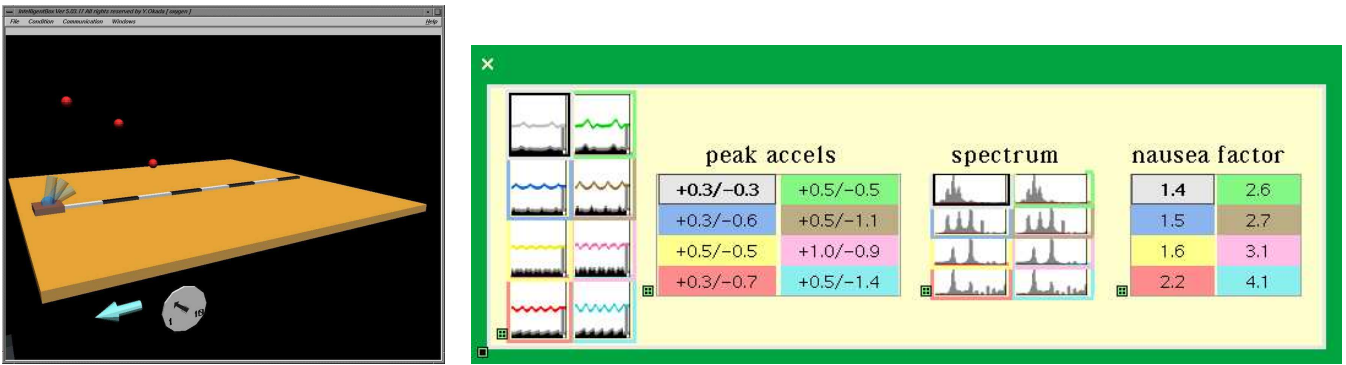


Figure 1: Left: An early example of a subjunctive interface, applied to a model of the flight of projectiles launched against a headwind. In the situation shown here, the user has chosen to create multiple scenarios based on different values for the angle of the launcher – specified by pressing a hot-key at various points during the direct-manipulation adjustment of that angle. As a result, three launchers now exist in parallel; the three balls seen here in mid-flight were launched simultaneously in the three scenarios a few seconds earlier. The user can now adjust wind speed (using the large arrow at bottom left) and projectile mass (the dial) and observe simultaneously how the launches from the different angles would be affected. If wanted, these parameters too can each be given multiple values; this will simply multiply the number of balls launched on each firing.

Right: A feedback tool for a vehicle designer, showing the simulated response of a suspension system running over a bumpy road. The designer has control over a number of variables in the suspension design (not shown in this figure), and is here examining how the current settings would perform in eight scenarios representing alternative road patterns and speeds. Each of the four response measurements (motion plot, acceleration values, frequency spectrum, and a motion-sickness measure) is showing dynamically updated values for each of the scenarios, arranged as ‘small multiples’ displays to help the user correlate the values between them. For example, the scenario at top left would subject passengers to accelerations of ± 0.3 , and has an estimated nausea factor of 1.4.

- The user should be able to control scenarios in parallel, for example by adjusting an input parameter that is shared by many scenarios and seeing instantaneously the effect of this adjustment on each scenario. (One could say that the cues available for reflection-in-action during such adjustment are thus multiplied by the number of scenarios being affected.)

Two examples of the many experimental subjunctive interfaces that have been built to date are shown in Figure 1. These are examples of design based on dynamic simulation. Subjunctive interfaces can also be useful for design of static artefacts – such as an architectural model that incorporates what-if elements, or a year planner that allows each future event to be scheduled for several alternative possible dates. In these latter domains the interface would help the user to work with the branching possibilities established by those alternatives, for example offering rich reflection-in-action when the user comes to add further elements and discovers what would be (in)consistent with the existing elements in each scenario.

As well as some user studies in Copenhagen demonstrating significant benefits from a subjunctive interface designed for browsing and comparing multi-attribute data (Lunzer & Hornbæk, to appear), other researchers have begun to obtain successful results from their own extensions of the theme – notably the recent work on supporting alternatives in image processing, under the banner Parallel Paths (Terry *et al.*, to appear).

Future: Ingredient-Based Computing

Figure 2 shows the first, simple example of a tool demonstrating Ingredient-Based Computing – a specialisation of subjunctive interfaces suited to applications that can be characterised as spreadsheet-like acyclic graphs of cells and formulas. This application model, which we believe can subsume a wide range of today’s applications (including, not least, spreadsheets), offers opportunities to simplify the handling of knock-on effects of alternative values that the user assigns to cells within the dependency graph.

One goal for ingredient-based computing is to cater for mixed-initiative variation – i.e., to allow alternatives to be generated not by direct user request but from automated processes. For example, computational processes that provide unsolicited information for user reflection (such as critics, context-sensing recommenders, or other forms of agent) may come up with conflicting values depending on the assumptions that each embodies. Rather than try to muddle all these elements together, it would make sense to replicate these scenarios automatically in a way that lets the user understand the various offerings – including how each has arisen – and make informed choices about which to attend to.

My hopes for this workshop include finding an opportunity to discuss concrete issues relating to such mixed-initiative support, and also relating to situations that would call for establishing long-lived alternatives – i.e., scenarios that remain separate, yet individually relevant, for weeks or months rather than just during the course of a single usage session.

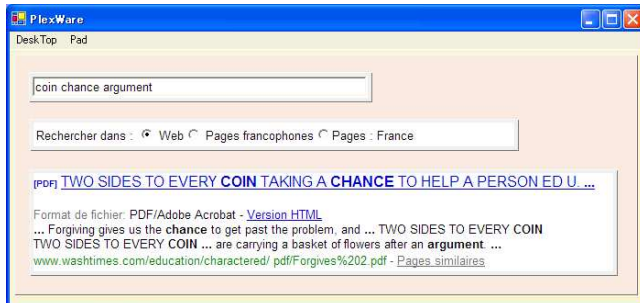


Figure 2: A trivial example of ingredient-based computing, showing the pursuit of multiple parallel navigations on the World Wide Web. On the left the user has set up a combination of cells in a tool called C3W (Clip, Connect and Clone for the Web) that is being developed in collaboration with Jun Fujima at Hokkaido University [paper submitted for review]. These cells contain active elements extracted from the search site google.fr, and hande, respectively, the search input field; the switch that selects whether the search should cover the whole Web, just pages in French, or just pages in France; and the top result (showing page name and keyword context) from a completed Google search. On the right the user has replicated the switch cell to create three alternatives, and has set each to a different option; because the search-result cell depends directly on the switch (through a pseudo-formula whose ‘calculation’ involves submitting a query to Google), it has been automatically replicated too. Entering new search keywords into the tool in this state will launch three queries, and display the three respective top results.

References

- Fischer, G. (2002) ‘Beyond “Couch Potatoes”: From Consumers to Designers and Active Contributors.’ *First Monday* 7(12), December 2002.
http://www.firstmonday.dk/issues/issue7_12/
- Hofstadter, D.R. (1979) ‘Gödel, Escher, Bach: an Eternal Golden Braid’. Basic Books.
- Lunzer, A. (1996) *Reconnaissance: a widely applicable approach encouraging well-informed choices in computer-based tasks*. Ph.D. thesis. TR-1996-4, Dept of Computing Science, Univ of Glasgow, Feb 1996. 266pp.
- Lunzer, A. (1998) ‘Towards the Subjunctive Interface: General Support for Parameter Exploration by Overlaying Alternative Application States.’ In *Late Breaking Hot Topics Proceedings of IEEE Visualization '98*, Research Triangle Park, North Carolina, Oct 1998, 45-48.
- 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 (INTERACT '99)*, Edinburgh, Scotland, Aug 1999, 474-482.
- 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 the Australian Conference on Computer-Human Interaction (OZCHI 2003)*, Brisbane, Australia, Dec 2003, 202-210.
- Lunzer, A. and Hornbæk, K. (to appear) ‘Usability Studies on a Visualisation for Parallel Display and Control of Alternative Scenarios.’ In *Proceedings of The 7th International Working Conference on Advanced Visual Interfaces (AVI 2004)*, Gallipoli, Italy, May 2004.
- Terry, M., Mynatt, E., Nakakoji, K. and Yamamoto, Y. (to appear) ‘Variation in Element and Action: Supporting Simultaneous Development of Alternative Solutions.’ In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*, Vienna, Austria, April 2004.