Design as Traversal and Consequences:
An Exploration Tool for Experimental Designs

Christopher G. Jennings*   Arthur E. Kirkpatrick†

Graphics, Visualization, and Usability Lab
Department of Computing Science
Simon Fraser University

Abstract

We present a design space explorer for the space of experimental designs. For many design problems, design decisions are determined by the consequences of the design rather than its elemental parts. To support this need, the explorer is constructed to make the designer aware of design-level options, provide a structured context for design, and provide feedback on the consequences of design decisions. We argue that this approach encourages the designer to consider a wider variety of designs, which will lead to more effective designs overall. In a qualitative study, experiment designers using the explorer were found to consider a wider variety of designs and more designs overall than they reported considering in their normal practice.

CR Categories: H.5.3 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces

Keywords: design spaces, design space exploration, history capture, design rationale, experimental design, creativity support

1   Introduction

Design is a process of considering options, and many times these options consist of tradeoffs, relaxing one constraint in order to tighten another. Such tradeoffs can be seen as a two-step process: deriving a new candidate design, often by modifying one that has already been considered, and then evaluating that new candidate in terms of the project goals. Typically, this process is informal, and the steps are not explicit. Although this informality makes the process fluid and permits ambiguities that can spark creativity, an informal process has costs as well. Designers can work from habit, making small modifications to stock designs. Like a 15-speed bicycle, their design process has gears that never get used. This retreat into habit is all the more likely when important consequences of a design are difficult to estimate. In such cases, it takes far less effort to work with known designs whose consequences have already been estimated, than to design *de novo* and incur the effort of estimation. A cheaper design process, but less effective design: stock designs provide only a loose solution to the specific problem.

To make it easier for designers to craft a solution specific to the problem, we have developed a system that explicitly represents these previously informal aspects of design. The system has three key principles:

1. It makes the designer aware of the available design options. Options are represented as moves that traverse an abstract

*cjennings@acm.org
†ted@sfu.ca

space.

2. It explicitly represents the underlying structure connecting designs. When the designer makes a move, varying an existing design to create a new one, the new design is displayed in context with the preceding designs. These designs remain displayed and available for comparison and to serve as the basis of other moves.

3. It provides fast, accurate feedback on the consequences of each design.

We argue that explicitly representing this process encourages designers to consider more candidate designs, to try a wider range of design options, and to think strategically, in terms of lines of inquiry rather than just individual designs. This diversity of designs and breadth of vision should in turn produce a final design more closely tailored to the specific problem.

The contribution of this system is twofold. The details of its interface and conceptual model demonstrate an approach to design support in domains amenable to formal description. In addition, the underlying ideas of traversal and consequence may be applicable in domains that do not have an inherently formal structure but have aspects that may be formalized.

We proceed by surveying previous work on design exploration and design choice. We then describe the domain of designs for analysis of variance (ANOVA) experiments. In Section 4 we discuss the typical processes of experiment designers, and prior empirical results about design processes in other fields. The discussion highlights the relatively small number of design alternatives that are typically considered. We then describe our system. Section 5 describes our representation of individual designs, which summarizes their key consequences to allow easy comparison with other designs. We next (Section 6) describe the representation of past designs as a traversal through a design space. Because this representation is fundamentally interactive, we describe it through an extended example based on a real design problem. In Section 7 we present results of an initial qualitative study in which five designers used the explorer. The final discussion (Section 8) considers the implications of the study and possible applications of the ideas of traversal and consequence to domains that lack a formal structure.

2   Related Work

Many papers have been written about the HCI aspects of supporting design. Here we will review that work which most directly addresses representing the design space and supporting design choice.

2.1   Representing the Design Space

Design space explorers (design explorers hereafter) are a computational model for supporting the design process. Built on a metaphor

of design as exploration [19], the design process is seen as a sequence of choices, with each choice moving the design to a new state. The approach draws strongly on Newell and Simon's work on problem-space navigation [13]. Design exploration is different from the puzzles considered in their work in that design does not have a formally-specified goal whose attainment can be recognized by an algorithm [3]. Due to this inherent ambiguity, designers make more contingent moves, backtrack more, and often work on multiple designs in parallel. Design explorers support these activities by providing history capture, multiple active design paths, and ongoing reminders of the available design moves.

Jones's review of design methods [8] divides design activities into three stages: *divergence*, in which the problem to be solved by the design is not completely defined, *transformation*, in which the general approach of the design is discovered, and *convergence*, in which the general approach is refined into the final design. Design exploration applies most directly to the transformation and convergence stages. Exploration is used to discover general approaches, while much of the convergence stage is automated by the computation of generative details. Consequences also play roles in both stages: first by supporting gross evaluations of the direction that trade-offs take under different approaches, and then to fine-tune those trade-offs once a specific approach has been chosen.

Existing design explorers have focused on the design of objects in space, such as two-dimensional room layouts [6] or solid shapes [7]. This reflects their basis in Stiny's work on shape grammars [16]. Consequently, their displays have featured the spatial arrangement of design elements. The abstract operators and constraints generating the design have typically remained implicit, without graphical representation.

Our system extends previous design explorer work by displaying the explored design space rather than the current design. It displays the track of all past designs, together with the design moves connecting them, and it arranges them according to their similarity to an initial design state. This approach explicitly presents the designer's task as one of trying alternate sequences of design moves, taken from an explicit, finite set of such moves. The designs themselves are displayed, in full or summarized form, as points in the space. These summaries characterize important consequences of the design rather than the design state or, as in an approach used by Stump et. al. [17], abstract indicators of the design state.

Stump et. al. [17] proposed an exploration model of design by shopping: the designer chooses designs that they like, and the system attempts to find optimal combinations of features. Optimization of this kind is in general not possible in experimental design. Changes to the design imply new assumptions about the research domain, and only an expert in that domain can determine which assumptions are valid for a given research question.

Previous design explorers emphasize reducing the relative cost of computing generative details, letting the designer focus on making broader changes [19]. How a circuit is laid out is unimportant so long as it is functionally correct and reasonably efficient. While our system does compute generative details of this kind, our focus is on reducing the cost of another kind of design activity, namely that of design evaluation. By this we mean evaluating the impact of different design approaches on one or more consequences.

In experimental design, the choices and consequences interact across the statistical and research domains. The designer must make decisions about which questions they really want to answer, how badly, and under what assumptions. These are research domain choices, but to compare them their consequences must be translated into and estimated in the statistical domain. Hence, experimental design consists largely of estimating the statistical implications of choices in the research domain. Computing these estimates is tedious. Our design explorer computes them for every design move, making it easier to compare a large number of designs.

2.2 Supporting Design Choice

Many projects have explored methods of supporting choice in design. Shneiderman describes a general framework for software support of any creative activity [15]. Of the eight activities he describes, design explorers provide support to three: exploring solutions, composing artifacts, and reviewing and replaying session histories. A particular strength of our design explorer is that it supports all three activities with a single mechanism, explicit navigation of the design space.

Terry et. al. [18] present an approach to supporting multiple simultaneous solutions to a design problem. Their system focuses on designs that cannot be well-described in terms of generative operators. Hence, they emphasize displaying multiple designs, which can be modified individually or as a group. In contrast, the design space considered in our work has stronger structure, and we emphasize displaying the space of designs created so far.

Design rationale systems are related to design explorers in that both focus on the design space. These systems are used to capture the reasoning behind design decisions along with the design changes. Providing a design rationale establishes an argument for the design that may be reviewed by critics or used as an aid in future design tasks [12]. In contrast, the purpose of design explorers is to support the current design task rather than future work. Although not a focus of this article, our system allows the association of informal design rationale annotations with both individual designs and explored spaces.

Klemmer et. al. [9] present the Designers' Outpost, a tool for designing web sites. This system combines history capture with support for informal design rationale annotations. The design history is navigated by scrolling through a linearized tree of thumbnails of the design state. In contrast to the history state approach, the design summaries we present feature consequences of the design state rather than a summary of the state itself. Because these summaries are often sufficient for decision-making, our system can remove the notion of an active or current moment in the design history. Instead, we present only the space of explored designs. Operations are performed directly on designs in the space, in any order, without first selecting a current design. The use of a zooming user interface (ZUI) [14] makes the full details of designs in the space available by zooming in when they are needed.

A common problem for systems that perform history capture is identifying which states are meaningful enough to be of historic importance [5]. This is not a significant issue in design explorers because they are defined in terms of operations, or moves, that transform one design directly into another design rather than relying on the aggregate effect of a sequence of lower-level editing operations. Consequently, the product of nearly every operation is expected to be worth capturing. In practice, the designer still occasionally wishes to group a few moves into a single result. In our explorer, this action is provided as a free side effect of the mechanism for performing operations on designs.

3 THE DOMAIN OF EXPERIMENTAL DESIGN

The design space that we model in our design explorer is classical experimental design for ANOVA designs. We will briefly discuss the factors that make experimental design a challenging task that is well-suited to design support.

3.1 The Value of Design Support

The domain of experimental design has three features that particularly recommend it for application of a design support tool.

First, it is a domain of substantial impact. ANOVA designs are widely used in HCI, the behavioural, basic, and clinical sciences,

government, and industry. Statistical consultants spend considerable time advising their clients on ANOVA designs.

Second, experimental design is both subtle and risky. It is difficult to get a design right because the mathematical model may miss important features of the research domain, small design changes can require surprisingly large changes to the analysis method, and evaluating designs requires making assumptions about the outcome of the experiment before data have been collected.

Third, the consequences that the designer must balance are generally difficult to compute by hand and hard to informally estimate—but many can be computed or estimated by machine.

Due to all of these factors, designers may well benefit from representing experimental design as a consequence-centered exploration of a space. By providing the designer with a high-level view of how the current design fits into the larger space of designs, she can begin to think of her design strategically. She will be encouraged to see the possible variants of the current design, to think in terms of the possible ways the design can be changed, and to more readily discover patterns of transformation that are likely to improve the design. She will tend to be more thorough and produce better designs: cheaper, more sensitive, more generalizable, or a combination. She will also have a clearer idea of the trade-offs that were made in the design, which will help her meet the needs of the experiment's sponsors.

3.2 Trade-offs in the Experimental Design Domain

All design problems require striking a balance between opposing trade-offs. In experimental design, there are six kinds of trade-offs:

1. The risk of drawing an incorrect conclusion from the results. This is represented in the statistical model by α (the risk of false positives) and β (the risk of false negatives).

2. The range of effect magnitudes which can be reliably detected (sensitivity). As the difference between two groups gets smaller, the designer must sacrifice other goals (such as low cost) to be able to find a statistically significant difference. Estimating the smallest effect magnitude that a design can reliably detect is called power analysis [4].

3. The generalizability of the results. The designer can make gains in other trade-offs by sacrificing how widely the results apply. (The designer might argue that the results still generalize to the larger population based on domain knowledge.)

4. The monetary cost of running the experiment. This is a significant concern in most research domains. The typical application of power analysis is to determine the minimum number of experimental units needed to find a difference of the expected magnitude: this ensures that money is not wasted taking unneeded measurements.

5. The structural complexity of the experiment. Complex designs have implications beyond the cost of taking the measurements. For example, changes that complicate the data gathering protocol increases the probability of accidentally—or intentionally—violating that protocol when performing the experiment.

6. How directly the dependent measure captures the true phenomenon of interest.

Lorenzen and Anderson provide an extended example of making these tradeoffs through six revisions of a design [11, Chap. 6]. In our design explorer, trade-off (1) is specified by the designer, while trade-offs (2)–(6) are consequences of the design. Our design explorer provides visualizations for trade-offs (2)–(4) in the design's summary representation (described in Section 5). Trade-off (5) is not presented directly, although it can be rapidly gauged by inspecting the design's detailed state representation. The interpretation of trade-off (6) is relative to the research question, so it cannot be computed automatically. However, a designer can assess trade-off (6) by comparing the tests available from the design with their research question. In addition, a constraint system (also described in Section 5) allows the designer to make statements about which tests are relevant to their research.

4 EXAMINING THE DESIGN PROCESS

Early in the development of our design explorer, we interviewed a statistical consultant to establish typical design practice in experimental design. His observations were later confirmed during detailed process interviews that we conducted as part of the user study described in Section 7. Here we use the collected observations from both sources to describe typical experimental design processes.

In practice, researchers tend not to consider a large number of designs. Inexperienced designers reported considering very few designs—typically only one or two—while experienced designers considered a few more. The interviews suggested five principal reasons for this:

First, some domains, such as the physical sciences, can be so constrained that few design choices are available. In such domains, exploring a space of designs may be less productive than in less restrictive domains. Computing design consequences is still valuable: for example, cost estimates can be used for budget allocation.

Second, because substantial work is needed to accurately estimate many important design consequences, researchers prefer to use rules of thumb to guide design decisions and avoid comparing multiple designs.

Third, many researchers neglect experimental design, leaving it to the last minute or even collecting data without any design. Once data is collected, the design is effectively established and little change is possible.

Fourth, many researchers are trained to design experiments by choosing the closest match from a list of predefined, named designs; there are only a few ways to modify such a design once it is selected.

Finally, although researchers usually have some training in the components of experimental design, few seem to have training in a design process. Rather, they develop an approach through trial and error. Even experienced designers tend to use a mish-mash of techniques. The typical approach starts with a design that worked in the past and has similar features, adjusts it based on rules of thumb, and then (sometimes) estimates the statistical power.

Considering all of these factors, experiment designers are especially likely to benefit from the principles introduced in Section 1: many designers have an incomplete understanding of the design operations available; there is currently no single widely-adopted approach to structuring the relationships between designs; and designers currently rely heavily on rules of thumb to estimate their design consequences.

More experienced designers reported considering more designs than less experienced ones, as well as considering a wider range of general approaches. Akin observed similar patterns in the design activities of architects [1]. He observed that experienced architects first considered a breadth of possible solutions and then investigated some of them in depth, whereas novices tended to immediately pursue their first idea and backtracked only when they ran into difficulty.

While considering a wider variety of designs is an important part of the transition to being an expert designer, it seems that even experienced experimental designers might try more approaches if the cost of estimating their design consequences were reduced. This

suggests that novice and experienced designers alike may benefit from our approach.

In the remainder of the paper, we describe how our design explorer applies the three guiding principles to support experimental design. The next section describes the design representation, which displays design options and estimated consequences. In Section 6, we describe the representation of design structure and demonstrate how all principles are used in developing an experimental design.

5 DESIGN REPRESENTATIONS

The design explorer's interface is a ZUI that uses two principal representations for designs at different zoom levels. When zoomed in, designs are presented in a table form that describes the current design state. When zoomed out, designs are presented as summaries of their consequences.

In the design state (zoomed-in) representation, the design is split into a matrix of rectangular cells. The first cell contains a design summary, and the remaining cells—up to 2^n of them, where n is number of main effects in the design—each describe a main effect, an (optional) interaction effect, or the error term. Figure 3 shows part of the detail view of a simple design (some of the cells have been left out to save space).

The design summary (zoomed-out) representation consists of 7 elements, of which 3 provide state information about the design and 4 represent design consequences (see Figure 4).

Each design is assigned a unique design number when it is added to the space. This provides a fixed visual anchor to aid in navigation and visual search, and captures chronology of the history tree. Designs also have an editable name. An initial name that describes the move(s) performed on the parent to generate the design is automatically assigned. This short name label can be supplemented with arbitrary text to record design rationale. These elements are identified in Figure 4 as (b), (g), and (h), respectively.

The remaining elements represent design consequences: cost, detectability, the inference space, and constraint satisfaction. These are identified in Figure 4 as (d), (e), (f), and (c), respectively.

The cost to run an experiment is important in most research domains. We estimate the cost of running an experiment using a probabilistic model. The designer estimates two kinds of costs for each main effect: a fixed cost to be paid for each experimental unit, and a change cost to be paid when one unit is switched for another. For example, paying each participant $10 would be a fixed cost, while providing a brand new set of headphones every time participants change would be a change cost. This change cost can be minimized by blocking participants (running all trials with one participant before switching to the next participant). Displaying costs shows one consequence of blocking an effect—there are also statistical implications, which will be shown in the detectability values.

Cost is indicated using a wedge; the taller end indicates higher cost. A red needle indicates the cost of the present design, relative to other explored designs. A blue zone indicates the area where the design is deemed too expensive, as determined by a user-defined constraint (described later).

The power of a test for an effect is estimated in terms of the spread of the means for each level of the effect. If the means are widely spread, they are more readily distinguished from the background variability. An effect's *detectability* is defined in terms of this spread. Informally, detectability is the narrowest spread of the effect that can be detected with a given risk of false positives and false negatives. Detectability is unitless, expressed in standard deviations of the error term used to test the effect. Precise definitions of detectability for fixed and random effects are provided by Lorenzen and Anderson [11, pp.104–108]. Detectability is similar to Cohen's d (for two levels) and f (for two or more levels) [4].

Compared to the standard approach to power analysis, detectability has the advantage that the designer does not need estimates of the standard deviations of the effects to use the system. However, it has the disadvantage that the designer's qualitative judgments about the design's power are less precise [10]. It suffices for our purposes, although standard power analysis values are easily obtained if the user can estimate the standard deviations.

Within a research domain, the interpretation of detectabilities is non-linear. For example, all detectabilities above 1 standard deviation may be considered equally bad, while the difference between 0.5 and 0.75 standard deviations may be considered substantial. To assist the designer in making fast qualitative judgments, a domain-specific function maps detectabilities to an abstract rating of zero to four "stars." This is presented in the design summary as a rectangle containing four Greek crosses. (We used Greek crosses rather than actual star shapes because the simpler design of the cross made it easier to compare mean ratings.) A needle indicates the mean star rating for all of the effects, while a gray region indicates the range over all effects. Individual detectability bars for each effect are available in the zoomed-in design representation.

The inference space of a design represents how widely the results can be generalized beyond the samples used in the experiment. As its use is outside the scope of this paper, and it was not used in the qualitative study (Section 7), we do not describe it further.

Some research domain-dependent consequences manifest in all research domains, yet vary in their specifics between domains. For example, the researcher may really only be interested in testing some of the effects in the design. We provide support for these kinds of domain-dependent trade-offs by allowing the user to describe them as global constraints. A pair of concentric circles in the upper-right corner of the design summary provides continual feedback on how designs meet the constraints on a space. When no constraints are met, neither circle is lit. If some constraints are met, the outer circle is lit (drawn in green). Meeting all constraints lights both circles.

6 DESIGN STRUCTURE: THE EXPLORER IN USE

We illustrate the design explorer's representation of design structure, and the use of all its features together, by presenting the design process for a small HCI experiment (see http://gruvi.cs.sfu.ca/videos/xds/ for a video). The explored space is presented as Figure 2. In the interest of brevity, this example does not exercise every feature of the system. Our primary interest will be balancing detectability against cost.

At all times, the interface displays the space of explored designs on a two-dimensional grid. At the start of a session, this consists of only the Empty Design, which acts as an origin for the space.

New designs are generated by performing moves on existing designs and are arranged automatically by similarity to their parents. The arrangement explicitly represents design as exploring a design space, and gives the designer a sense of how designs are structurally related by observing their spatial relationship. Parent designs are linked to their children by an arrow, and children are always placed close to their parent. This ensures a clear path for the designer to follow when reviewing the design history.

The experiment is a 3-way factorial design with three effects that we will call A (with four levels), B (two levels), and C (two levels). The participants of the study also constitute a fourth, random, effect, that will begin with 8 levels. As is typical in HCI studies, we are concerned with the number of participants (which affects the power of the study) but not with their effect size (the variation between individual participants). The design process will focus on the consequences of varying the number and arrangement of the participants.

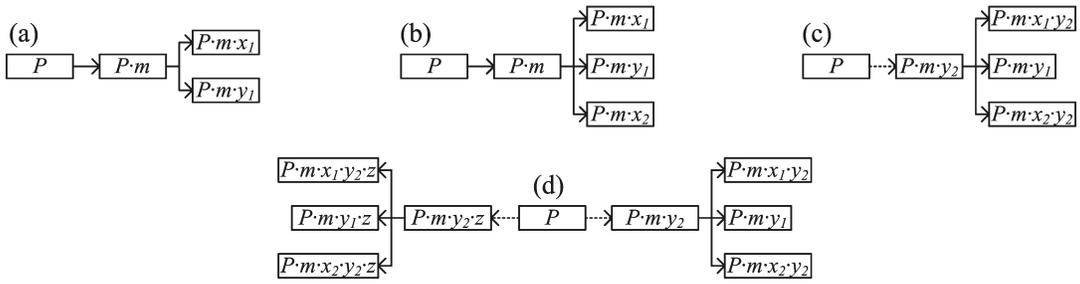To begin, we create a base design by adding the four effects. This

Figure 1: The application of moves in the design space: (a) an initial space featuring a parent design (P) and children formed by composing P with moves; (b) applying x_2 to $P \cdot m$ as a *shallow* move creates a new branch ($P \cdot m \cdot x_2$) in the history tree; (c) applying y_2 to $P \cdot m$ as a *deep* move recursively applies the move in place—$P \cdot m \cdot y_1$ is unaffected because y_1 overrides the newly inherited y_2 (the dotted line indicates the application of multiple moves); (d) applying z to $P \cdot m \cdot y_2$ as a *branch* move replicates the $P \cdot m \cdot y_2$ branch, applying z deeply to the result.

means performing a sequence of four moves (one for each effect) on the Empty Design. To perform a move, the mouse pointer is moved over the target design, causing the design to be surrounded by a halo of buttons representing the possible moves. When a move is selected, a dialog window appears over the halo button and the move parameters are specified. The move is then performed, and the explored space is updated to display the result.

As we are creating a base design rather than exploring, we will use the variant *deep* application for these moves. The usual *shallow* application creates a new branch in the history tree, adding a new design to the explored space. In contrast, deep moves *replace* the target design with the new one. In addition, children of the target design will recursively inherit the change entailed by the move unless they override the change with a conflicting move. The design exploration model is responsible for adjusting these designs as needed to ensure that they remain valid and consistent. Figure 1 illustrates the ways in which moves can be applied.

Having set up our base design with deep moves, we are ready to begin exploring. Our main research interest is in effect A. We want the experiment to detect differences in A as small as 0.5 standard deviations, so we add a global constraint that designs will only be acceptable if A's detectability is 0.5 or less. We are not as interested in effects B and C, so we set their constraints slightly higher (0.6).

The base design (Design 0) meets none of the constraints, so further refinements are necessary. We start by increasing the number of participants to 12. This is done as a shallow move application, so a new design branch is created to display the result (Design 1).

At this point our attention is interrupted—we realize that we may have mistakenly given A only two levels in the base design. We verify the mistake by zooming in to the detailed representation and checking the number of levels of A. To correct it, we apply a deep move to Design 0 that increases A's levels to four. Unlike a traditional undo mechanism, the deep application propagates the correction to Design 1 since it is a child of Design 0 and does not override the number of effect levels. After verifying the correction, we zoom back out and resume exploration.

Getting back to our new design branch, we find that increasing to 12 participants brings the detectability of B and C to .51, so the outer rim of the green light is lit, but the center remains off because the constraint for A is unsatisfied. The new design is also more expensive, as we would expect, but the cost bar clearly indicates the magnitude of the increase. Increasing participants to 16 (Design 2) improves the detectability (and increases cost) further. This design completely fills the green light, indicating a design with the necessary sensitivity for every effect.

The designs considered so far have all been within-subjects. We wonder how this approach would compare to a between-subjects design that engaged some collaborators to perform each level of the

A effect at a different site (Design 3). This requires making several concurrent changes to the design. We perform the first of these as a shallow move to start a new branch, then perform the remaining moves as deep moves on the new design. The dotted connecting line to Design 3 indicates that it is a product of multiple moves.

With only the 8 participants inherited from the base design, the detectability is the worst of any design so far, as shown by the low range of values in the detectability bar. As before, we increase the number of participants to 12 (Design 4) and then 16 (Design 5), but neither has sufficient power. We could add even more participants, but the cost of running 16 is already higher than we would like.

Instead, we decide to find out what we might gain if we accepted a higher risk of erroneous results. Increasing α to .10 and β to .20, we find that the detectability for all effects finally falls within the desired range. The between-subjects design seems distinctly worse than the within-subjects design: even given 16 participants, we must accept substantially higher risk to get our desired detectability. We will probably abandon the between-subjects approach.

Our attention returns to the within-subjects branch. The last design in that branch has the sensitivity we want, but it is rather expensive. We wonder if we can find a way to use the cheaper 12 participant version (Design 1). Pointing the mouse at this design, its move halo appears. On scanning the available moves, we are reminded that we just solved a similar problem in the between-subjects line by accepting a higher risk of erroneous results.

Operating on Design 1, we try relaxing just the β value. The result (Design 7) is observed to have the same detectability as the 16 participant design, but the same cost as its parent.

This analysis allows us to make an informed trade-off of cost and risk for this design. Whereas rules of thumb only give general directions (increasing participants always reduces the risk), we now have estimates of how much expense we will have to pay to reduce our level of risk a fixed amount.

7 QUALITATIVE STUDY

We conducted a qualitative study to assess how our design explorer affected the design process of experiment designers.

Five participants took part in the study. These were chosen to represent a variety of research domains and previous experimental design experience. The domains included academic HCI research, quality control and product testing in the food industry, pharmacoeconomics, and health care research. Two of the participants were highly experienced with experimental design using ANOVA experiments; one of them teaches a graduate-level experimental design course. Two of the participants were novices with a basic statistical background but little practical experience. The last was at

an intermediate level, having constructed many designs in his own domain but lacking the breadth of experience of the experts.

7.1 Methods

Participants took part in a single study session consisting of three parts and lasting about 90–120 minutes. In the first part, participants were asked about the kinds of experiments they perform and about their experimental design process. This background was later used to discuss explorer concepts with familiar terminology, and to build an example design. The length of each interview varied, while the remaining two parts were alloted roughly 30 minutes each.

In the second part, participants were given some training on the system. The interviewer worked with the participant to set up a basic design based on a typical problem in the participant's practice. During this process, the interviewer explained the basics of using moves to change the design, how to use the ZUI, and what the consequence summaries represent. The initial design was set up using only deep moves. At the end, a single shallow move was performed to show the participant how the software would let them create new designs without "forgetting" older ones.

In the third part, the design explorer was turned over to the participants, who worked with the system using a think-aloud protocol. No particular course of action was suggested; rather, participants were simply instructed to try the system out. No other details on the theory of the design explorer's operation were provided, although additional program features were sometimes described in response to questions or actions from the participant. Although any usability issues that we observed were noted, our focus during this session was on observing the design process adopted by the participant while using the explorer.

As we wanted to observe how participants would work with multiple designs, the interviewer was prepared to inject occasional prompts like, "what do you think would happen if you tried doing x to this design?" In practice, this was unnecessary: all of the participants explored the space enthusiastically without prompting.

7.2 Results

Confirming our hypothesis, the participants created many alternative designs. Three participants extended the co-developed example for the entire session. The other two developed fresh designs starting from an empty design space. Of the second group, one developed a design based on an experiment they were planning, while the other created an abstract design that was used to experiment by trying various moves to observe the consequences.

All of the participants developed and tested hypotheses about design consequences at some point. In addition, all of the participants but one used the consequence visualizations to work on a realistic design problem. All four of them developed a design that was closer to the trade-offs they wanted.

Some of the typical comments made during exploration indicated this hypothesis testing process: "Let's see how much this will hurt the power." — "I like the power on this one, but I want to make the cost go down." — "OK, so now if I change α, will that affect the cost?" — "How can I get this effect to have a test?" — "It doesn't seem to make much difference to make something random if you only have two effects."

Once they were comfortable using the basic design moves, all but one participant asked the interviewer to explain some of the moves that had not been demonstrated in the second part of the session. The participant that did not ask was one of the experts, and so may have already grasped their purpose. Typically, after asking about one of these "advanced" moves, the participant would immediately try to find a way to apply it.

All of the participants tried most of the moves available at some point in the session, and everyone tried at least one of the "ad-vanced" moves. The major exception was the move to statistically nest an effect, which only one person tried but three people asked about. However, nesting is arguably the most difficult move to understand, rarely taught during introductorystatistics courses, and infrequently used in practice.

All but one of the participants learned to make appropriate use of deep moves to hide interim design changes. One participant (an expert) chose shallow moves almost exclusively, occasionally commenting, "I know I could use a deep move here, but I'm interested to see what happens." All of the participants used the design consequence summaries. No one asked for the detectability or cost visualizations to be explained again, although many people asked technical questions about how they are computed.

Although the focus of our study was on the use of moves and the consequence visualizations rather than the layout of the design space, we did notice that out of about 150 designs that the participants collectively generated, there were only 3 instances where they appeared visibly confused about a navigation task. We also observed several instances where the participant appeared to quickly navigate to far off branches by first visually searching back through the design history to locate their destination. The spatial arrangement does not appear to significantly hinder performance.

Overall, we were pleased by how many designs participants explored, and by how readily experimented with unfamiliar moves and sought to apply both the designs and the moves to real problems. Most importantly, they did not restrict themselves to incremental refinements. They tried a wider range of general approaches, and more designs overall, than they reported using in normal practice. This confirms the value of our three design principles for promoting the consideration of additional designs, enabling the discovery of more creative designs overall. Furthermore, the four participants who attempted to improve an initial design succeeded. Since the tool led them to discover a novel, beneficial solution, we conclude that it is effective at supporting creativity.

We temper these conclusions with some caution. After all, the participants were using a novel interface to work on hypothetical designs: mistakes had no consequences. They might revert to more familiar methods if they were working on real designs.

Yet even within the restricted context of one hour of working with the explorer, some users learned techniques that might change their future work. Three users spontaneously reported leaving the session with a better understanding of the consequences of their design choices. One of the expert users reported that in the past he had used fixed effects instead of random effects because they gave better sensitivity, despite a reduction in the generality of the results (a rule of thumb). On comparing both approaches during the session, he concluded that (for his design domain) the difference was smaller than he thought, and that he would likely try using random effects in his next design as a result. These statements suggest that even this brief exposure to our design explorer changed how some participants will design experiments in the future. We suggest that more pronounced changes would occur if they were actually using the explorer when designing their next studies.

8 DISCUSSION

Our design explorer implements our key principles for the field of experimental design, and initial results show that actual designers using it considered more designs than usual, and were able to improve their designs. We now consider two kinds of generalizations of this work: Application of our explorer to actual experimental design, and application of our principles to other domains.

Generating more designs is neither necessary nor sufficient for producing better designs. However, if a better design exists, it must be generated to be discovered. Our system allows the designer to quickly generate and compare alternatives, and it focuses on pre-

senting the measures that the designer needs to judge the outcome. This increases designer efficiency and reduces cognitive burden by externalizing those properties of the design. As a result, the designer is left free to concentrate on higher-level tasks, such as conceiving of alternative approaches. Designers who use our system to make a thoughtful exploration of the design space are more likely to find new approaches and develop final designs that are better tailored to the specific problem if one exists, or to confidently conclude that finding a significantly improved design is unlikely.

One criticism that has been aimed at previous design explorers is that they only solve simplified toy problems [2]. By contrast, our explorer is not a toy: it features a complete representation of a design domain which—as we discussed in Section 3—has substantial real-world impact.

We have concerns about the scalability of consequence displays. The display algorithms scale easily, supporting good interactive performance for spaces of 1,000 designs. We are less confident of the usefulness of such a display. The problem never arose in our study, with participants generating at most 25 designs. We intend to study displaying larger numbers of design consequences in future work. Making such a consequence display useful will likely require careful design.

Many interesting design domains are not well-suited to a design explorer implementation: they may be too ambiguous to construct an effective model, or the moves of interest may be too dependent on the specific task. Although we use a design space exploration approach to implement the principles of displaying design options and representing design structure, other tools might use other approaches. Alternatively, design support might be restricted to a well-defined subset of design tasks. As a practical example, consider the refactoring tools provided by many software development environments. Software design is not generally well-suited to a design space representation, but refactoring tools are effective at supporting variant production by automating small-scale, concrete design moves which are tedious and error-prone to perform by hand.

Providing fast, accurate feedback on design consequences should be effective independently of the other two principles. The use of consequences to motivate design choices could be applied to many design domains that lack the structure needed for a design space representation.

Finding good design consequences can present its own difficulties. Clearly, the appropriate consequences must be evaluated separately for each domain based on the factors that influence design decisions. A tool that supports web site design might make use of consequences such as download time, word count, the reading difficulty of the text, the complexity of the navigation graph, or its suitability for disabled persons using a screen reader.

For some domains, it may not be possible to evaluate the most useful consequences automatically: extracting a summary of the plot of a story, for example. Sometimes it is possible to approximate the consequence or to provide the consequence in a simplified form that misses some details but is still useful in guiding decisions; otherwise, those consequence must be abandoned. In the latter case, it still may be worth supporting second tier consequences: the result is still better guidance for making design decisions.

9   Conclusion

A design space explorer based on three design support principles (make the designer aware of the design options; provide a structured context for design; provide fast, accurate feedback on the design-level consequences of decisions) can encourage the designer to consider more designs, and a wider variety of designs overall. Our design explorer incorporates these principles using a persistent "halo" of moves that directly transform one design into another, explicit traversal of the design space, and summarizing design states as visualizations of their consequences.

Design has a powerful effect on daily life, opening some possibilities and restricting others. The more closely a design is tailored to its specific requirements, the better its fit to the activities of its users. Our design explorer shows promise for encouraging such tailored designs in the domain of ANOVA experiments. We hope that others will adapt, extend, and improve the techniques presented here to make a positive impact in other design domains.

10   Acknowledgements

References

[1] Ömer Akin. Psychology of the early design process. In *Design Decision Support Systems Conference*, 1994.

[2] Ömer Akin. The whittled design space. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 20(2):83–88, 2006.

[3] Ömer Akin and Cem Akin. On the process of creativity in puzzles, inventions, and designs. *Automation in Construction*, 7(2–3):123–138, 1998.

[4] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum, Hillsdale, NJ, 2nd edition, 1988.

[5] Tracy Hammond and Krzysztof Gajos. An agent-based system for capturing and indexing software design meetings. Presented at the International Workshop On Agents in Design, August 2002.

[6] Mikako Harada, Andrew Witkin, and David Baraff. Interactive physically-based manipulation of discrete/continuous models. In *SIGGRAPH 1995*, pages 199–208, 1995.

[7] Jeff Heisserman. Generative geometric design. *IEEE Computer Graphics and Applications*, 14(2):37–45, 1994.

[8] J. Christopher Jones. *Design Methods*. Van Nostrand Reinhold, New York, 2nd edition, 1992.

[9] Scott R. Klemmer, Michael Thomsen, Ethan Phelps-Goodman, Robert Lee, and James A. Landay. Where do web sites come from?: Capturing and interacting with design history. In *CHI 2002*, pages 1–8. ACM Press, 2002.

[10] Russel V. Lenth. Some practical guidelines for effective sample size determination. *The American Statistician*, 55:187–193, 2001.

[11] Thomas J. Lorenzen and Virgil L. Anderson. *Design of Experiments: A No-Name Approach*, volume 139 of *Statistics: Textbooks and Monographs*. Marcel Dekker, Inc., New York, 1993.

[12] T. P. Moran and J. M. Carroll. *Design Rationale: Concepts, Techniques, and Use*. Lawrence Erlbaum Associates, 1996.

[13] Allen Newell and Herbert A. Simon. *Human Problem Solving*. Prentice Hall, Englewood Cliffs, NJ, 1972.

[14] Ken Perlin and David Fox. Pad — An alternative approach to the computer interface. In *SIGGRAPH*, pages 57–64, 1993.

[15] Ben Shneiderman. Creating creativity: User interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction*, 7(1):114–138, 2000.

[16] George Stiny. Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, 7(3):343–352, 1980.

[17] Gary M. Stump, Mike Yukish, Timothy W. Simpson, and E. Nathan Harris. Design space visualization and its application to a design by shopping paradigm. In *Design Engineering Technical Conferences*, 2003.

[18] Michael Terry, Elizabeth D. Mynatt, Kumiyo Nakakoji, and Yasuhiro Yamamoto. Variation in element and action: Simultaneous development of alternative solutions. In *CHI 2004*, pages 711–718. ACM Press, 2004.

[19] Robert F. Woodbury and Andrew L. Burrow. Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 20(2):63–82, 2006.
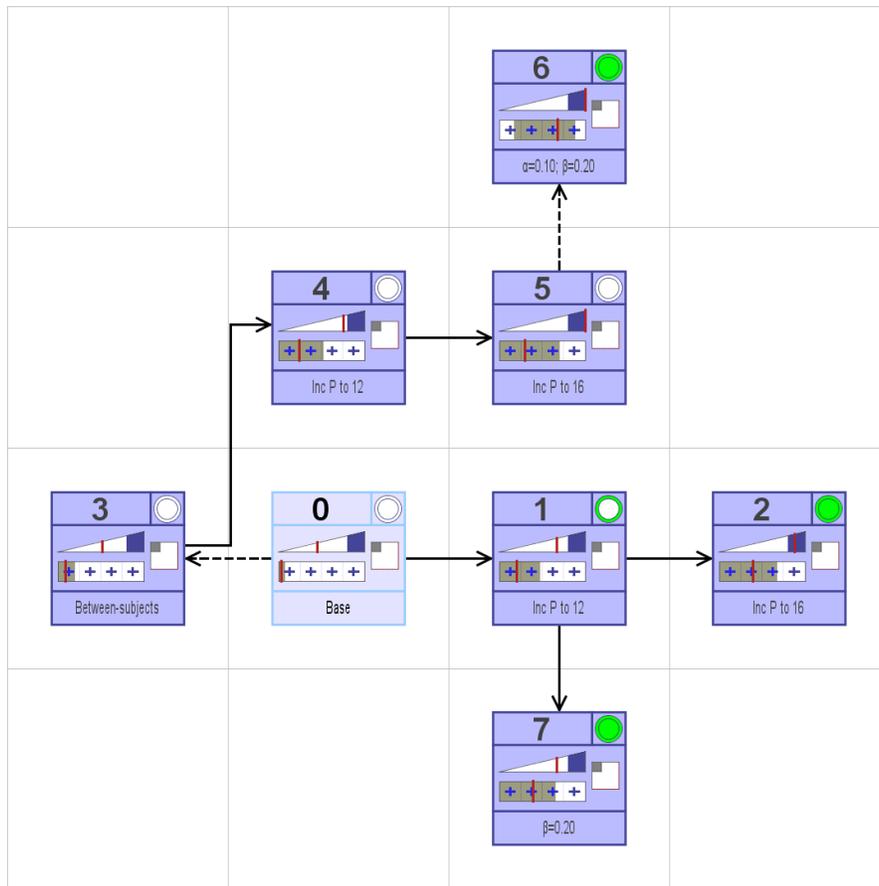
Figure 2: An example design space exploration session. The session is recounted in Section 6.
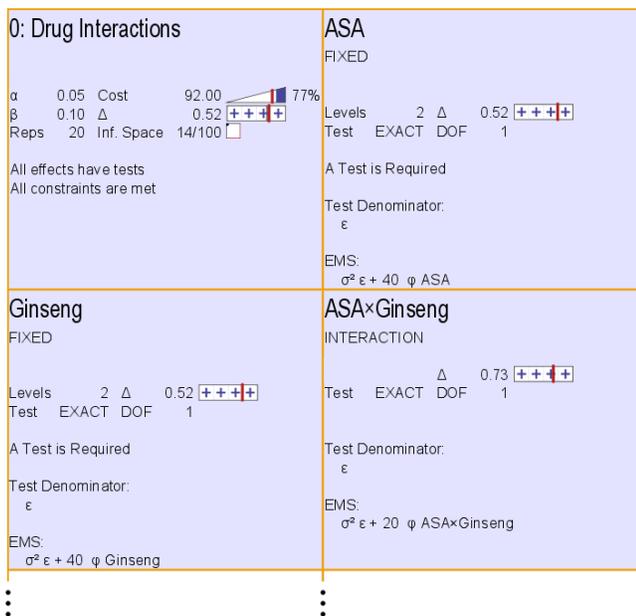

Figure 3: Part of the detailed representation of a design. The summary representation of the same design is presented in Figure 4.
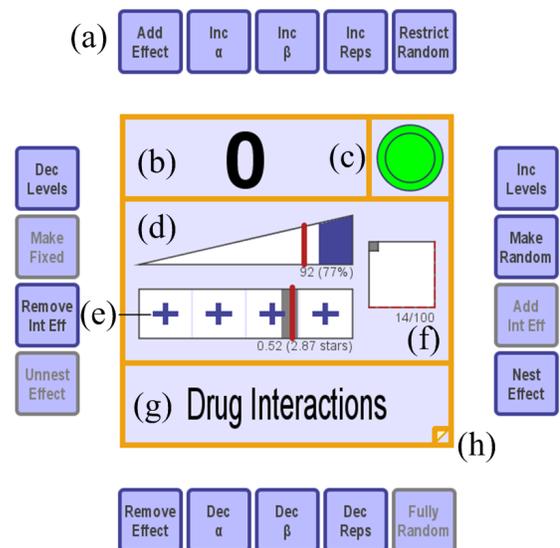

Figure 4: A design summary, showing: (a) halo buttons, presenting the possible moves; (b) design number; (c) constraint greenlight; (d) cost; (e) detectability; (f) inference space; (g) design name; (h) design rationale annotation indicator.