
Sketchplorer: A Mixed-Initiative Tool for Sketching and Exploring Interactive Layout Designs

Kashyap Todi^{1,2}

kashyap.todi@gmail.com

Daryl Weir¹

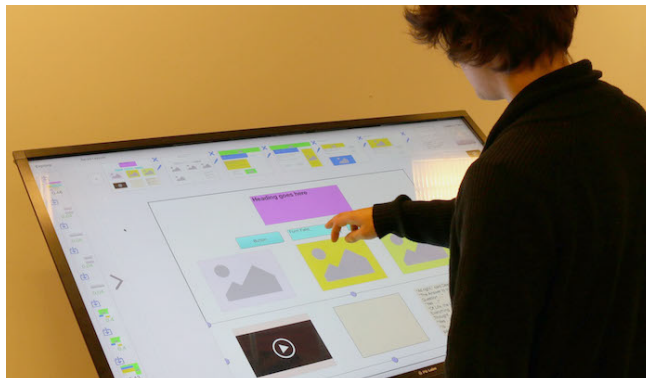
darylweir@gmail.com

Antti Oulasvirta¹

antti.oulasvirta@aalto.fi

¹Aalto University, Finland

²Hasselt University, Belgium



To cite this work, please refer to the original publication [4] in *Proceedings of the ACM SIGCHI Conference on Designing Interactive Systems (DIS 2016)*.
Copyright © 2017 for this paper is held by the author(s).
Proceedings of MICI 2017: CHI Workshop on Mixed-Initiative Creative Interfaces.

Abstract

This workshop paper discusses our mixed-initiative approach that enables designers to rapidly *sketch* and *explore* interactive layout designs. Although optimisation methods can attack very complex design problems, their insistence on precise objectives and a point optimum is a poor fit with sketching practices. Typical optimisation tools also fail to incorporate the human in the loop. *Sketchplorer* is a mixed-initiative sketching tool that uses a real-time layout optimiser. It automatically infers the designer's task to search for both local improvements to the current design and global (radical) alternatives. Using predictive models of sensorimotor performance and perception, it generates suggestions that interactively steer the designer towards more usable and aesthetic layouts without overriding them or demanding extensive input.

While this position paper summarises our work from the mixed-initiative perspective, further details can be found in the original publication [4].

Author Keywords

Sketching; Model-based optimisation; Visual Layouts; Mixed-Initiative; Design Tools

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces

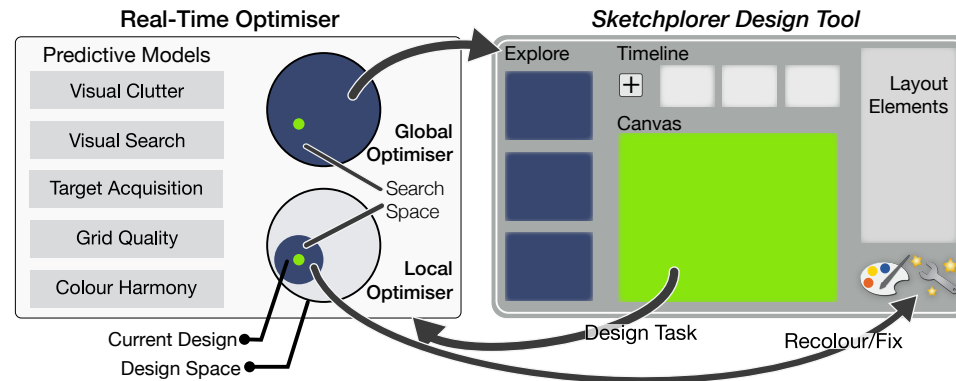


Figure 1: *Sketchplorer* is a mixed-initiative interactive sketching tool that uses a novel approach to real-time layout optimisation. While the designer sketches, the optimiser infers the design task and searches for local improvements (*recolour*; *fix*) and global alternatives (*explore*). It uses predictive models of performance and perception to generate new designs and suggests these to the designer.

Introduction

Sketching is a powerful visual thinking tool, employed by designers to explore different solutions to a given design problem. It is a very fluid process, and tends to be ambiguous and uncertain [2]. Optimisation, on the other hand, can efficiently attack very complex design problems, and systematically improve both usability and aesthetics of designs. Traditionally, optimisation is a rigid process, and demands *precise* input. These contradicting requirements inhibit designers from employing powerful optimisation techniques during sketching and ideation phases of design.

Sketchploration is a novel concept that brings real-time optimisation to visual design tools. It couples a designer and an optimiser in a new “serendipitous” way, supporting the natural flow of sketching and ideation. Technically, sketchploration (Figure 1) extends *model-based interface optimisation* [1, 3] to real-time design exploration under ill-specified and changing design goals. It relaxes the re-

quirements for controlling an optimiser. As the designer sketches, the optimiser infers the design task automatically. It then searches for both local (small changes) and global (large changes) alternatives. Importantly, our optimiser deploys several predictive models of user performance and perception, adapted from literature. This allows it to make informed suggestions that “pull” designers toward *usable* and *aesthetic* designs.

Sketchplorer is a mixed-initiative design tool that showcases the first investigation of this concept, focusing on *interactive layouts* familiar from GUIs, web pages, menus, and dialogs. It provides designers with a multitouch environment for sketching, and for exploring interactive layout designs generated computationally.

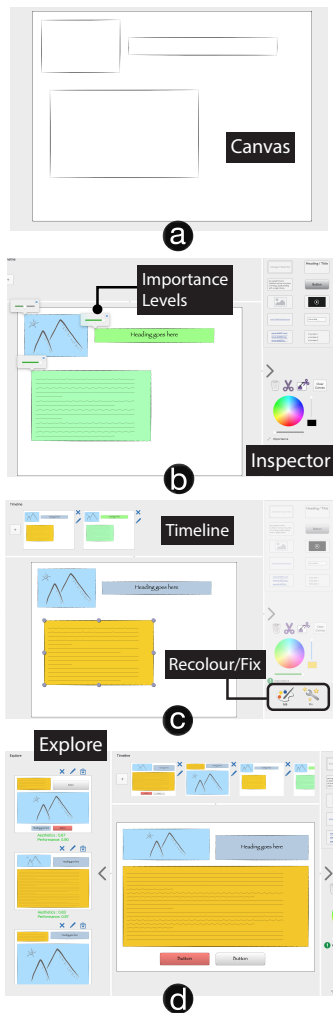


Figure 2: A Walkthrough. The designer (a) ambiguously sketches layout designs, and (b) refines and add details. They have immediate access to a timeline of saved designs. The optimiser suggests (c) **local** (*fix and recolour*) changes, and (d) **global** (*explore*) design alternatives.

Walkthrough: Designing a Blog Page

This walkthrough illustrates the use of Sketchplorer, our design tool based on the sketchploration concept, from a designer's perspective.

Sketching the initial layout: Sketchplorer initially presents the designer with an empty canvas. The designers starts sketching by creating a structure for her design (Figure 2a). Drawn elements can be moved around, or resized, at any time. She ambiguously sketches out boxes, serving as proxies for the elements of her blog page. Sketchplorer infers the design task, and the optimiser starts computing both local and global suggestions in the background.

Refining and adding details: An *inspector* panel sits on the right-edge of the display, and can be pulled out at any time. This can be used to specify details such as element type, colour, and importance (Figure 2b). The designer now takes a moment to add these details to some of the elements. For instance, she indicates that her page has a heading and a paragraph element, and marks them as being of high importance. Satisfied with the first version of the design, she taps the save ('+') button. This adds the current design to the designer's *timeline*, and provides a preview (Figure 2c). Saved designs can be retrieved, and edited, at any later time.

Making local changes (fix and recolour): While the designer sketches and refines, the system continuously streams the description of the current design to the optimiser. The local optimiser uses the current design as a starting point to suggest *fixes* and *recoloured* designs. Pulsating icons in the inspector panel indicate the availability of these suggestions (Figure 2c). The designer now refers to these suggestions, and realises that by using a recolouring suggestion, she can make the paragraph of text stand out. She selects this, and continues working on the sketch.

Exploring global alternatives: By abstracting from the current design, the global optimiser retrieves unique designs, and returns them to the designer. An *explore* panel, residing on the left edge of the display (Figure 2d), is periodically updated with these designs. The designer browses through the list of alternatives, and finds two interesting alternatives. She adds the first to her list of saved designs, and drags the second onto the canvas, to continue working on it.

In a short duration, the designer's collection of *saved layouts* is populated with several feasible alternatives—some sketched by the designer, and the others with the aid of the optimiser. While the above phases appear to be linear, in practice, sketching and exploration phases are intertwined.

Sketchploration: Mixed-Initiative Layout Design

As is evident from the workflow, Sketchplorer relies heavily on both the designer and the machine to iterate over layout designs. In our work, we identified that such layout design tasks require both—human and machine knowledge—for successful completion of the task in a short timespan. The mixed-initiative nature of the tool is critical to its success.

Designers are good at quickly sketching out a design task, and at using their internal design knowledge to identify feasible designs. On the other hand, computational methods can explore a large number of solutions in a short amount of time. By employing predictive models of human performance and aesthetics, they can optimise for certain criteria, and objectively improve designs for the user population. However, since machines do not have complete knowledge of the design tasks, and might not be aware of aspects such as semantics of the layout, we rely on designers to finally evaluate the suggestions, select feasible solutions, and optionally refine them further. By employing this ap-

proach, we can enable rapid as well as systematic exploration of the design space, to arrive at feasible solutions for the given design task.

Sketchplorer divides the task of creating layout designs as follows:

Designer:

1. Specify the design task by sketching on the canvas.
2. Select feasible solutions from a larger set of suggestions.
3. Further iterate over these suggestions to arrive at one or several final designs.

Optimiser:

1. Interpret the design task from the sketch.
2. Exploit and explore the design space to generate suggestions.
3. Periodically return a set of feasible designs that improve objective aspects of the design.

To fluidly support these sketching and exploration activities, Sketchplorer provides designers with a set of interactions, as highlighted in the following section.

Overview of Interactions

Sketchplorer is designed for a multitouch environment using a large display, and uses touch gestures for all controls.

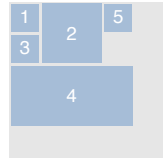
Sketching and Refinement: Sketchplorer allows designers to either sketch ambiguous bounding boxes for layout elements, or pick out a specific element type and place it on the canvas. It takes care to accurately order every element on the canvas, without designer intervention. Each time an element is changed, the hierarchy is inferred, and the ordering dynamically adapted. Hierarchical groups of elements

can be selected and manipulated at the same time. This inferred hierarchy is also essential for the internal representation of a layout in the optimiser. A colour picker allows designers to select the hue–saturation combination, and adjust the brightness. Double-tapping on an element reveals an in-place pop-over, and allows adding details without having to move to the inspector. An element’s importance can be adjusted in the inspector panel. Alternatively, an overlay can be enabled, that displays the importance of every element, and allows batch adjustments (Figure 2b).

Saved versions and timeline: In Sketchplorer, the current design can be added to the timeline at any time. This timeline provides an overview of all saved designs. Designers can non-linearly continue iterating over any saved design by dragging them back to the canvas. Thus, the timeline allows comparing of designs, provides an overview of the evolution of designs, and enables iteration over sketches.

Interacting with the optimiser: To the designer, the optimisation appears as a two-pronged approach, consisting of local and global optimisation. The *local optimiser* observes every change in the design. It suggests recolouring and fixes that maintain the overall composition of the original sketch, but improve certain aspects. It also provides harmonious recolouring suggestions, which improve aesthetics and performance. In contrast, the *global optimiser* listens exclusively to changes in design tasks, and acts upon them. It abstracts away from exact details, allowing it to explore the entire design space, and generate unique designs. It performs exploration in real-time and periodically returns improved results, displayed to the designer in the expandable *explore* panel. The global optimiser focuses on creating unique and improved solutions, not on refining or polishing a solution to make them perfect.

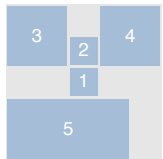
As the optimiser generates these suggestions, they are



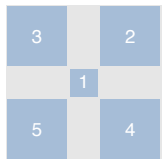
Visual clutter



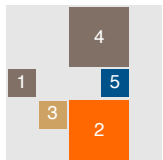
Visual search



Target acquisition



Grid quality



Colour harmony

revealed to the designer in the tool (Figure 1). Designers can cycle through local improvements (recolour or fixes) by tapping on the respective icons. Global improvements (exploratory suggestions) appear as thumbnails, and the designer can scroll through them, save select designs to the timeline, refine them further on the canvas, or reject them.

Predictive Models and Objective Function

Whereas previous work has used heuristics and focused on visual search and motor performance [1, 3], Sketchplorer addresses spatial and colour aspects of layouts. To achieve this, we use the following predictive models:

1. **Visual Clutter:** Rosenholtz Model
2. **Visual Search:** Kieras-Hornof Model
3. **Target Acquisition:** Fitts' Law
4. **Grid Quality:** Balinsky's Symmetry Measure
5. **Colour Harmony:** Cohen's Harmonic Colour Sets

Figure 3 illustrates optimisation results for individual objectives, for a sample task with five elements. We define a multi-objective task where we seek to minimise a weighted combination of the outputs of these models:

$$U = \sum_{i=1}^5 w_i S_i, \quad (1)$$

where the weights w_i sum to 1 and the individual objectives S_i are normalised to the range $[0, 1]$.

Evaluation

We evaluated our model-based optimisation approach through a user study, which compared participants' performance with a baseline design to that with an optimised designs. Quantitative results favoured our approach. Further details can be found in our full paper.

To evaluate Sketchplorer as a design tool, we conducted a study with experienced designers, where they were given sample design tasks, and asked to create multiple feasible designs. We found that designers referred to both local and global suggestions while creating their designs. 9 out of 10 designers took advantage of optimised suggestions, to achieve their final designs. We also observed that by using these design suggestions, designers were able to create a larger number of feasible results in a short amount of time. This is especially encouraging since the purpose of sketching designs is to explore the feasible space, to detect a variety of promising solutions.

Lessons for Mixed-Initiative Creative Interfaces

Sketchplorer is a system that enables designers to rapidly generate several design ideas for a given layout design problem, and employs predictive models and optimisation techniques to assist designers in their task. During the design and implementation of our system, we learnt some valuable lessons that we feel can benefit other mixed-initiative tools. Some of these are listed as follows:

1. **Minimise Control Requirements:** Sketchplorer infers the design task by abstracting from designers' concrete sketches. It does not force designers to explicitly specify the task, or additional details, and allows them to focus on the sketching task. Although additional details can help the optimiser find better solutions, they distract from the primary objectives of the user, and hence are not desirable.
2. **Appropriate Division of Labour:** The system does not attempt to replace the designer. Instead, the designer and the computer complement each others' skills. While the designer is good at recognising good designs and understanding semantics of a layout,

Figure 3: Outcomes when optimising for a single objective. Sketchplorer combines these into multi-objective search.

the computer can efficiently generate several good solutions that objectively improve performance and aesthetic criteria. Thus, the system rapidly generates several solutions that it deems to be objectively good, and relies upon the designer to select a subset of these that semantically make the most sense in fulfilling the design requirements.

3. **Support Local and Global Improvements:** Sketchplorer attempts to both exploit and explore the design space to improve designs. Designers can thus get support to make minor improvements as well as major redesigns. This approach can thus be employed to generate new solutions at early stages, as well as fine-tune existing solutions.
4. **Provide Quick and Plenty Solutions:** While the designer sketches out an idea, the system continually analyses the sketches and searches for improvements. Since it does not have a complete understanding of the design task, it tries to generate several new solutions, and returns them to the designer. By presenting them as a non-disruptive overview, it allows designers to quickly glance through them and discard undesirable solutions, and pick out one or more suitable designs.

Conclusion

By integrating real-time optimisation into a sketching tool, *sketchploration* opens up opportunities for designers to efficiently explore alternative designs, and improve the usability and aesthetics of outcome designs. Our mixed-initiative

approach to interface design enables the designer and optimiser to complement each other, to arrive at good solutions in a short span of time. We have conducted user studies to validate our optimisation technique, and the interactions supported by Sketchplorer. The results from these are positive, but a full discussion is outside the scope of this abstract, and we invite readers to refer to [4] for further details.

References

- [1] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: Interactive Optimization of Menu Systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 331–342. DOI:<http://dx.doi.org/10.1145/2501988.2502024>
- [2] Bill Buxton. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [3] Krzysztof Gajos and Daniel S. Weld. 2004. SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces (IUI '04)*. ACM, New York, NY, USA, 93–100. DOI:<http://dx.doi.org/10.1145/964442.964461>
- [4] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 543–555. DOI:<http://dx.doi.org/10.1145/2901790.2901817>