

SNARF Taxonomies for Sketching Novel and Realistic Functionality

Christopher Drexelius & Rohit Uday Talwalkar & Brian Cho & Brian P. Bailey

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{drexeliu, talwalka, bcho2, bpbailey}@cs.uiuc.edu

Abstract

Paper prototyping is often used during the critical stage of early design. While paper prototyping requires a small time investment, it can fail to accurately convey the application's behavior. Existing computerized tools allow designers to express both appearance and interactions through sketching; designers may even execute their sketches. However, these tools have limitations, such as scalability and an inability to represent some complex behaviors.

By asking users to sketch interfaces that are complex and novel, we developed a family of sketching taxonomies. We also created a set of suggested design notations based on the specific behaviors and characteristics users sketched. Furthermore, we developed an iterative user-centric approach that can be used to expand and refine the SNARF taxonomies and suggested notations. Most users, given these suggested notations were able to double their sketched behavior. This result suggests our notations would be a welcome addition to any sketch-based tool. We also identified the potential of color as a linking mechanism.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [User Interfaces]: Prototyping, User-centered Design

1. Introduction

In the early stages of design, developers need to effectively communicate their ideas to colleagues. Effective communication is necessary to make good decisions. Poorly made decisions may require time-consuming fixes in later stages of development.

Interface designers traditionally explore early design ideas on paper [GD96]. Paper provides rich affordances that can greatly benefit the designer [CB04]. However, communicating the behavior of an interface on paper is difficult [RSI96]. Design tools such as Macromedia's Authorware [Rie99] offer designers the option of using computers to communicate detailed behavior, but at significant cost in terms of time and effort required when compared to paper prototyping [BK03].

Existing tools of this sort require high investment because they do not support the natural expressiveness afforded by sketching. Sketches allow quick exploration of ideas. Researchers have developed informal computer design tools SILK [LM95], DENIM [LNHL00], and DEMAIS [BKC01a] that have been relatively successful in combining the benefits of sketching with the capabilities provided by a computer. Using an informal tool, designers can sketch interfaces that contain behaviors. Upon completion, the sketches may then be executed to explore the specified behaviors.

Yet there are limitations to existing tools which must be resolved. These tools often do not scale to larger designs and have difficulty creating complex interactions. We believe the tools have these limitations because they were derived from simple design tasks or existing design artifacts. By exploring

a more complicated task domain, our study is able to extend the functionality offered by existing tools and sketching languages. And by employing a user-centric approach, we were able to develop recommendations which are highly effective and easy to use. Our user-centric iterative method also produces taxonomies, which can be used to compare and extend existing tools.

The rest of the paper is organized into the following sections: Contributions, Related Work, Method, Results, SNARF Family of Taxonomies and Suggested Notations, Lessons Learned, and Conclusion and Future Work.

2. Contributions

Our study led us to four noteworthy contributions regarding sketching and early design. First, we identified the SNARF taxonomies, which serve as a comparative base between existing sketching languages and provide possible areas of extension. Secondly, after our first user visit where users completed design tasks using free-form sketching, we developed a set of suggested sketching notations, which we refined in the next iteration. We also identified an iterative approach to construct the taxonomies and suggested notations, beginning with free-form sketching and recursing repeatedly with a set of evolving suggested notations. Finally, we found color to be an effective state linking mechanism.

2.1. SNARF Family of Taxonomies

These taxonomies represent the functionality users want to sketch. An iterative approach, involving user visits was required to capture relevant information. The three taxonomies we present in the SNARF Family of Taxonomies and Suggested Notations section are content, behavior, and state. This set of taxonomies adheres to a sentence metaphor where nouns, verbs, and adjectives correspond to content, behavior, and state respectively. Thus, the expressiveness of a sketching language can be judged by the number of sentences that can be formed with it. An expressive language allows designers more freedom in the designs they explore. In addition, this metric can be used in conjunction with our suggested notations to extend existing sketching languages.

2.2. Suggested Notations

Upon completion of the first round of our study, we identified a suggested set of notations based on user input. Upon completion of the second set of user visits, we were able to refine and evaluate our suggested notations. We also identified additional advanced notations, which we will evaluate in future work. Participant feedback presented in our results validates the usefulness of our suggested set of notations. The sketching notations are presented in detail in the SNARF Family of Taxonomies and Suggested Notations section.

2.3. Iterative User-Centric Method

The key to our study is employing a user-centric iterative method which produces, validates, and refines potential sketching notations based on user feedback. While user-centric methods are commonly used to develop interfaces, they are not commonly employed when developing a sketching language. Our method contrasts directly with traditional sketching language and tool development; the two approaches commonly employed when identifying the functionality needed for a sketching language are: examine existing artifacts for notations commonly used or infer commonly sketched behaviors through brainstorming.

Each iteration consists of a user study, classification of results, and refinement of the language. Participants either sketch free-form drawings or utilize an evolving set of suggested notations. Study participants define any additional desired behavior not supported by given notations. User-defined notations help to refine the taxonomies and extend the set of suggested notations.

2.4. Color-based State Transitions

Participants in our study commonly utilized color to distinguish between elements during the first round. For example, many users outlined each button within their design with a different color, visually indicating a unique result when the button is clicked. Traditionally, sketch-based design tools represent links of this type with arrows. However, arrow transitions do not scale well since they tend to clutter sketched designs, potentially confusing the designer and hindering productivity. In addition, arrows may not be suitable for large displays where related elements are distant from one another. On a partitioned display, it may be impossible to draw an arrow across a partition [BCR*03]. In addition to physical inconvenience, moving between distant portions of a large display may take a large amount of time. Finally, in a collaborative environment, arrows may even be impossible to use. Designers using current tools may be discouraged from fully exploring their ideas during early design for exactly these reasons.

Color linking has been used to clarify the visualization of complex data sets [Him98]. We envision a system that similarly takes advantage of color as links. Consider selecting the color blue from a color palette and drawing a button, followed by creating a storyboard of the same color. Clicking the button will show the linked storyboard. To keep the designer from getting lost, the system would have options to show and hide arrows corresponding to specified color links. Links using arrows will clutter the design space, but colors do not have the same cluttering effect. Since humans can only discern between a small set of colors, methods should be employed to carefully pick the colors used [Hea96]. Thus, the addition of color links can help overcome some of the known problems of arrow links.

3. Related Work

3.1. Design Practice Advancement

Processes and tools to improve design practice through computers have been developed through various methods of study. Wagner envisioned work in interface design augmented by computers through her own personal experience [Wag90]. Surveys of user interface designers conducted by Landay led to SILK [LM95]. Interviews conducted by Newman and Landay with professional web designers resulted in DENIM [NL00]. Bailey conducted interviews and surveys to study design practice, and based his language on design artifacts [BKC01b].

Additional methods have been used to further understand the design process. Lugt set up experimental meetings in his effort to explore the functions of sketching in design idea generation [vdL02]. Herbsleb and Kuwana analyzed the content of real design meetings to understand what knowledge is essential to designers [HK93]. Using the understanding of design process as a baseline, our research attempts to find required features of a design language. In contrast to the methods employed in previous work, our approach is to perform multiple iterations of controlled design sessions, thus creating a more accurate view of early design.

3.2. Informal Design Tool Implementation

Researchers have developed informal design tools that utilize sketching such as SILK, DENIM, and DEMAIS. These tools allow designers to quickly outline the design of an interface and execute that design. However, there are limitations to many current tools, which may lead to designer frustration and prevent the full realization of their conceptual design. Some features such as animation and advanced media content may not be present, and managing complexity for large designs may be difficult.

A simple design with few interactions can be quickly sketched into an executable format that is easy to understand. However, sketching complex and elaborate designs may be a long process that results in complicated and cluttered specifications. For example, a simple DVD menu in DEMAIS would require at least as many storyboards and arrows as menu items since storyboards are the only way to represent different states.

In addition, complicated interactions such as animation or variables are difficult to represent in each of these systems. For example, sketching a smooth animation of a person performing a cartwheel is not supported by SILK, DENIM, or DEMAIS. A method for adding animation capabilities in DEMAIS has been attempted; however, the approach requires the use of a scripting language [Hah04]. A purely sketch-based animation language has been developed, but this language is not coupled with a general-purpose informal sketching language [Ste].

We believe one reason that these tools exhibit shortcomings is that the sketching language each tool uses is based on a small sample of tasks limited to a specific domain (e.g. web-site development for DENIM, multimedia design for DEMAIS). This limited domain sampling produced languages that can easily support simple designs but may not support complex interactions necessary to represent novel design tasks. Development of the sketching language within DEMAIS stemmed from examination of existing artifacts, only accounting for design tasks attempted in practice. Our study asked users to complete a wider range of novel tasks in order to find characteristics needed in an effective sketching language. Furthermore, our iterative approach yields additional information which may be used to extend existing tools.

3.3. Informal Design Tool Guidelines

Plimmer and Apperley outline two general requirements for a digital sketch interface: usability and functionality [PA04]. Newman and Landay propose guidelines for potentially effective computer-based design tools [NL00]. A new design tool would have to

1. Use an informal user interface
2. Support multiple representations
3. Focus on early design phases
4. Integrate with other tools
5. Manage history and variations

Our study produces a family of taxonomies and suggested notations based on user sketches. The taxonomies and notations benefit existing tools by offering possible extensions and allow for comparative analysis. A truly powerful informal design tool would not only follow the guidelines identified by Newman and Landay, but also contain many identified notations that correspond to elements within our family of taxonomies, since they model behaviors users wish to sketch during early design.

4. Method

Our study consisted of two rounds: the first encouraged free-form creative sketching, and the second round evaluated our set of suggested notations. We based our notation set on results from the first round. Participants performed two sketching tasks each round, participated in both rounds, and had a different task set for each round. Each round lasted less than forty-five minutes.

Any additional iterations of our study will follow a similar procedure to the second round user study and analysis.

4.1. Users

Our user base consisted of fifteen students from the University of Illinois Urbana-Champaign; four of them were fe-

male. All our users except one of the females was a Computer Science graduate student. All users participated without compensation aside from refreshments.

4.2. Round One

The study consisted of a brief introduction and two design tasks, each followed by a survey. Users were instructed to sketch the appearance as well as the underlying behavior of the associated interface. A maximum of twelve minutes was given to complete each task. Despite the complex nature of the tasks, a large majority of users finished in the time given. Upon completion, participants explained the created design and completed a survey.

The materials provided to users include: large sheets of paper, pencils, erasers, colored markers, transparencies, dry-erase markers, multicolored post-it notes, scissors, and tape.

Four users performed task set A, four users performed task set B, and six users performed task set C; the tasks were divided into task sets below. For each task set, a randomly selected half of the users performed the tasks in the order listed, and the remaining half performed them in reverse order.

From the artifacts users sketched and verbal and textual descriptions of desired behaviors, we created a suggested set of notations. In situations where users did not sketch functionality, but verbally expressed functionality desires, we attempted to create a representative sketching notation. In addition, we began to create our family of taxonomies.

4.3. Round Two

The second round was similar to the first. However, in this round users were given a set of suggested sketching notations we produced from analysis of first round results and asked to complete their assigned design tasks using them. Along with the instructions from the first round, users were also given an introduction to the sketching notations, which took approximately ten minutes. While sketching, users had the option of adding extra notations that they wanted to define, which they could then use in their design. After users completed a task, we interpreted the resulting sketch and asked participants for clarification as needed. Finally, we asked users to complete a survey.

Each user was given a different task set from the first round; a user who performed task set A in the first study would perform task set B, etc.

Second round results were used to evaluate the usefulness of the suggested sketching notations. Users defined more advanced notations beyond our set of suggested notations. Based on user feedback, we created additional suggested notations and extended our taxonomies to include these desired behaviors.

4.4. Design Tasks

The following six tasks constitute our task set and were used in both rounds. Each task listed involves sketching complex interactions, animations, and relationships which are not supported by current informal tools. However, the tasks were designed to allow users to finish in a timely manner. Tasks are grouped by the task set to which they belong. For each task, a brief description is provided, along with the motivation for selecting that task.

4.4.1. Task Set A

- Global Navigation – Create three navigation bars for a video that is playing. One bar will change the time for the video stream, one bar will change the time for the audio stream, and the last bar will change the time for both streams and set them both to the same time. This task centers on synchronizing behavior of two scrollbars with a single global scrollbar. This type of relationship (a control which works upon another control) is not represented well in current informal tools.
- E-mail Client – Create an e-mail client interface composed of two panes. The top pane is scrollable and contains e-mail identification information for twenty e-mails at a time. When a user clicks on an e-mail entry, detailed information including all header fields (subject, from, to, etc.) and content are displayed in the bottom pane. This task focuses on linking interactions in one area of a sketch to another area. Behaviors such as indicating the two scrollbars and the corresponding scrollable areas are difficult using current tools.

4.4.2. Task Set B

- Water Cycle – Create an animation showing how water evaporates, condenses into clouds, and returns to earth as rain. During the animation, three short sound clips need to play in order to explain the process. This task forces users to focus on animation and timeline-based events. While some informal sketching tools do address animation, this task may be difficult using these tools due to its large scale and timeline-based events.
- Simple Machine – Create a visual representation of an interface where users simulate dropping an anvil onto a seesaw. Lifting the anvil is accomplished by clicking and dragging the anvil, and releasing the mouse button effectively "drops" the anvil. If the anvil strikes the empty side of the seesaw, a bowling ball on the other side is catapulted into the air. This task requires complicated state or conditional behaviors to be indicated in the sketch. In addition, the task requires certain objects to behave as "drag-drop-and-animate" objects, a feature not commonly supported by current tools.

4.4.3. Task Set C

- Campus Map – Create an interactive map of the University of Illinois campus with five to seven landmarks. Basic

information about a landmark appears when the cursor is placed over it, and clicking on a landmark shows users more detailed information. This task introduces scalability issues since it requires a large amount of links spread across the sketch. The task also includes multimedia content and advanced interactions.

- DVD Menu – Create a DVD menu, which includes play, special features, and cast menus. Clicking on each option displays subsequent screens containing more detailed information about the film. While a user is browsing each menu, a repeating background animation plays. This task addresses scalability issues; the more menu items and containers the user draws, the more cluttered their design may become. The DVD menu task also introduces issues such as variables and their influence on other objects' behaviors. For example, clicking "subtitles on" and then "play" should start the movie with subtitles displaying.

5. Results

Upon completion of each design task, participants rated two statements with a 7-point Likert Scale and a few free-response questions. We also performed a comparative analysis between the first and second rounds regarding sketched behaviors. Finally, we produced a taxonomy and suggested sketching notations, which can aid in developing a new sketching language or extending existing tools. Details of the SNARF taxonomy and related sketching notation suggestions are presented in depth in the next section.

5.1. Round One

Users indicated favorably that they would sketch the task ($\mu=5.89$, $\sigma=1.75$) and sketching helped communicate their designs ($\mu=6.18$, $\sigma=1.31$). These results validated the design tasks we selected and reinforced sketching as a valid approach during early design. In addition, we asked users if they knew of any existing tools or notations which could have aided them in their task. Users who responded to this question mentioned code/script based prototyping tools or existing design tools such as Authorware, DEMAIS, etc.

5.2. Round Two

Participants found the sketching notations useful ($\mu=5.86$, $\sigma=0.80$) and said they would use a tool containing similar notations ($\mu=5.64$, $\sigma=1.47$). The survey results validated the usefulness of our sketching language suggestions. These results also indicate individuals with minimal experience using existing sketch-based design tools would be interested in using a tool supporting our suggested notations.

Users were asked to define additional sketching notations required to complete their design tasks. This question addresses the completeness of our SNARF taxonomies, the

limits of our sketching notations, and exposes unexplored areas of the problem domain. Users defining new notations generally did so in order to address limitations with variables and parameter passing. Some participants also identified new notations relating to media, including pause, fast forward, rewind, upon completion, etc. Some users also re-defined correlation between draggable objects to formalize this behavior. Complex creational methods such as templates for numbered lists were also defined in order to duplicate unchanged portions of sketches while updating some numbered values for each entry; this notation simulates the behavior of a for loop.

The second question asked users to identify design tasks they felt the current sketching notations did not support. Users identified complex state management (variables and parameter passing) and perceived scalability limitations. These results will guide design task development for future study iterations.

Task	Round 1 Average	Round 2 Average
Global Navigation	6.25 (3.95)	12.5 (3.67)
E-mail Client	5.75 (3.1)	5.5 (2.17)
Water Cycle	9.75 (4.03)	18.6 (3.65)
Simple Machine	14 (14.14)	11.6 (3.05)
Campus Map	5.67 (4.41)	22.75 (9.91)
DVD Menu	9.83 (7.7)	24.25 (9.29)

Table 1: Average sketched behavior per task (Std. Dev)

We counted the total number of behaviors sketched in both rounds by counting only behaviors that were apparent from the sketches themselves; interactions defined with text and verbal explanations were not counted. Sketching behavior counts are averaged for each task and separated by round in Table 1.

These results clearly show that users sketched more behaviors in the second round. The first round of the simple machine task was heavily influenced by an outlier, removal of which produces an average consistent with the other results.

Users found the suggested sketching notations expressive enough to handle the complex behaviors required by their assigned tasks. For the e-mail task, users sketched the appearance of an e-mail client with which they had experience; they generally did not find it necessary to indicate the behaviors since a solution to this design task was familiar. However, the other tasks were sufficiently novel that participants produced a wider range of behaviors and designs.

6. SNARF Family of Taxonomies and Suggested Notations

Currently, three different SNARF taxonomies exist: content, behavior, and state. Our taxonomies can be used to iden-

tify additional notations that may be added to existing tools. Adding these notations may allow existing tools to overcome their limitations. For each category, we present the notations it contains. Since our suggested notations are iteratively generated and tested, notations we have not tested such as scaling animations are not presented.

Our taxonomy family adheres to the following metaphor: content, behavior, and state represent nouns, verbs, and adjectives respectively. Elements from each category are combined together to form a sentence. The more complex the vocabulary of a sketching language, the more design tasks it can accurately describe. We present an example sketch to further explain this metaphor and our notations.

6.1. Taxonomy Family Explanation

Each notation is prefaced by the justification for developing it based on the first round results. Our notations are presented in a similar format as given to users for the second round of our study. First, an explanation of the notation is given, followed by an example sketch which illustrates how to use the notation. Since most example sketches contain other notations, examples make more sense after reading all notations.

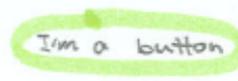
6.1.1. Content Taxonomy

The content category is composed of two sub-categories: recognized and non-recognized. Recognized objects fall into the following three categories: media, containers, and buttons. The media category contains audio, video, pictures, etc. Containers include storyboards, overlays, modal dialogs, etc. Any non-container content should be sketched within a container object to guarantee it is correctly recognized. For example, a button sketched outside a storyboard may be recognized incorrectly as a storyboard. The non-recognized category contains all sketches not contained within the recognized category. Since non-recognized content is not part of any sketching language, suggested notations do not apply.

1. **Storyboard:** A large rectangle (or an entire sheet of paper) represents a storyboard.
To create a storyboard: Draw a large rectangle that does not overlap with other content.
2. **Media Symbols:** Users indicated they wanted to show media playing with a symbol. We suggest using iconic symbols which users can drag instead of sketch, due to its complexity.
To add a video symbol: Place the video symbol in the container where video will play; if inside a button, plays when button is clicked.
To add an audio symbol: Place the audio symbol in the container where audio will play; if inside a button, plays when button is clicked.
3. **Button:** Users frequently drew bounded regions which also had an associated input and action such as "click to close".

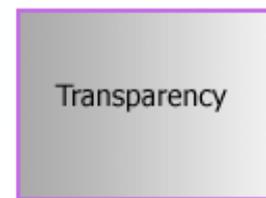
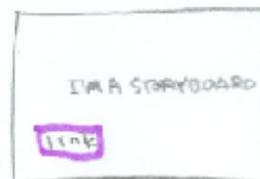


To create a button: Select a color and draw the outline of a closed region.



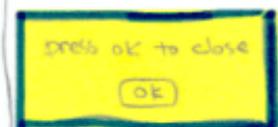
4. **Transparent Overlay:** Another container frequently used was a transparent overlay which added something to the storyboard but preserved parts of the existing sketch's behavior.

To create a transparent overlay: Attach a transparency. Any content visible through the transparency will have its behavior preserved.



5. **Modal Dialog:** Some users indicated a modal dialog for confirmation or warnings in their sketches with a post-it note.

To create a modal dialog: Attach a yellow post-it note.



6.1.2. Behavior Taxonomy

The behavior taxonomy consists of four sub-categories: media-specific, state updates, general, and animation.

Media-specific behaviors include play, stop, rewind, fast forward, pause, etc. These behaviors only apply to media content such as audio and video. Suggested notations for this category include:

1. **Play Media:** Users always drew a play button to indicate the media start action in their sketches.

When enclosed in a button, media plays when the button is clicked. When enclosed inside a storyboard (but not a button), media plays when the storyboard is shown.

2. **Stop Media:** Users always drew a stop button to indicate the media stop action in their sketches. A trigger may be added from a button to stop a media clip. To create a stop media link: (1) Create the trigger button. (2) Add a video or audio symbol. (3) Sketch an arrow with a stop media symbol as illustrated in the examples from the button to the media symbol.



3. **Repeat Media:** Some users indicated repeating media in the DVD menu with similar symbols. We adapted their designs to this notation. To repeat a media clip: Draw the repeat media symbol as shown below on the desired media icon.



State updates involve the interaction of a variety of content and variables and are the subject of future work.

Unlike the other behavior categories, general behavior may be applied to any content. General behavior consists of visibility changes and animations. Visibility changes are composed of fading in/out, showing, hiding, etc. Suggested notations for visibility changes include:

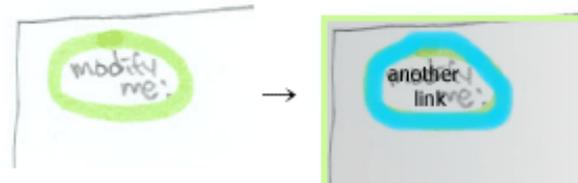
1. **Hover-Text/Mouseovers:** Users wanted mouseover actions to show elements such as text. To create hover-text/mouseovers: Within a button region, add hover text content with the same color as the button outline. Audio/video outlined in the same color as its containing button will start with a mouseover.



2. **Link:** Users frequently drew objects which changed the storyboard to a different one. To create a link: (1) Determine the color of the trigger button. (2) Outline the linked object with that color.



3. **Sub-Overlay Change:** Some participants used transparencies to indicate a partial change in a lower layer such as a button changing where it is linked to. This behavior only applies to transparent layer containers. To disable lower-level content: In the upper-level transparency, place a delete symbol over the lower-level content that should be disabled. To modify lower-level button links: Sketch a new button outline (changing the button link color) on the upper-level overlay.



The animation category consists of rotational, hierarchical, scaling, and spatial animations. Since rotational and scaling animations were identified at the conclusion of the second round, suggested notations for these categories do not yet exist. Suggested notations for the other two animation types are:

1. **Draggable-Object:** Users often used a cut-out or a post-it and verbally indicated its property as "draggable". To create a free-form draggable object: Cut a blue post-it and place it in a storyboard. To create a draggable object on a line: (1) Draw a line segment. (2) Cut a blue post-it and place it on the line.



2. **Correlated Draggable-Objects:** Users expressed a need for a shortcut to create two or more hierarchically related draggable objects.

To correlate draggable objects: Sketch a line between the two draggable objects.

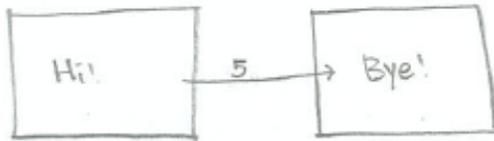
3. **Scrollbar:** Another shortcut users wanted was a scrollbar notation. This notation is a combination of draggable object, variable, and correlated draggable object.

To create a scrollbar: (1) Sketch the outline of a scrollable section. (2) Construct a scrollbar as shown below.



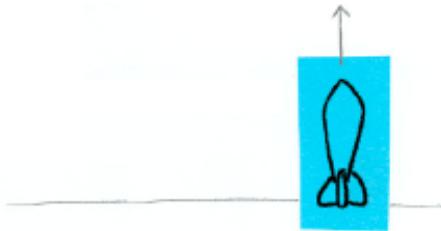
4. **Storyboard Transition:** Users expressed the need to control the transition from storyboard to storyboard during the DVD menu task.

To create a transition between storyboards: Sketch an arrow from one storyboard to another with a numerical value (in seconds).



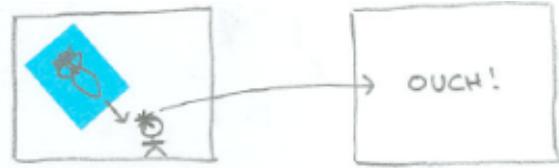
5. **Animated Draggable Object:** Some users depicted complex interactions with draggable objects. This notation allows the user to define the motion path of the draggable object.

To animate a draggable object: Sketch an arrow from the draggable object in the direction of desired motion.



6. **Collision:** For some tasks, users specified an action would take place when two objects overlapped each other, which is a collision.

To make a collision interaction: (1) Sketch a star on a static scene object as a transition trigger. (2) Sketch an arrow from the star to another storyboard.



7. **Basic Animation:** Users tried to express the parameters (such as motion path) for their animations.

(1) Cut an orange post-it and place in a storyboard. (2) Sketch an arrow indicating a path of motion for the animated object.



6.1.3. State Taxonomy

The state taxonomy is left as future work. We believe it will include such concepts as variables, conditional logic, and other such associations.

6.2. Example Design Sketch

We present a sketch in Figure 1 that an individual using our suggested notations might create in order to complete our DVD design task.

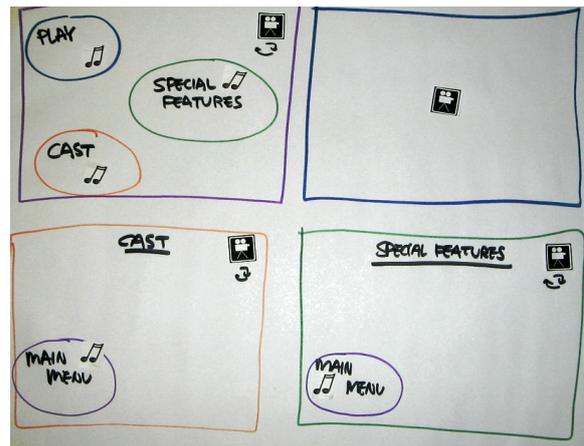


Figure 1: DVD interface designed using our sketching language.

During execution, the large colored square areas are interpreted as storyboards. Most black strokes such as the text, aside from the repeat symbols, are not recognized. The circular enclosed regions are interpreted as buttons since they are within a container. Clicking on any one of these buttons while executing the design sketch displays the storyboard outlined in the same color. Each button contains an audio icon indicating that an audio clip is played when the button is clicked. Video icons are present in each storyboard to indicate background animations and the feature video itself. Since they are present in the storyboard with no trigger notations, a new video begins playing whenever a storyboard transition occurs. Repeat notations are associated with the nearest video icon. This simple example highlights the expressive power of our sketching notations.

This example embodies our sentence metaphor as follows: play the movie when the “play” button is clicked. In this example, movie and button are nouns, representing content, and play is a verb, representing behavior. A more advanced example utilizing the state taxonomy might appear as: play the subtitled movie when the play button is clicked if subtitles have been turned on. In this case, the state, represented by the adjective “subtitled” and conditional logic, modifies the content being played.

7. Lessons Learned

7.1. Users will sketch more if they are given a set of sketchable building blocks.

In the second round, users generally were more inclined to use sketching notations for things they would normally have annotated by text in the first round. This means that if they know of a sketching notation that models some aspect of the current task, they will use it.

7.2. Novel designs foster large variations in the approaches taken by users.

In both rounds the novel designs had larger variation in how they were drawn. For tasks which were familiar to the user such as e-mail, the variation in the designs was smaller. This lesson stresses the importance of flexibility within taxonomies.

7.3. Users have difficulty sketching exact shapes such as media icons.

Users expressed some dissent about having to draw overly complicated icons or strokes. To solve this situation, sketching tools should incorporate draggable iconic representations of the sketches as a shortcut for users.

7.4. Users will limit their interface designs by their experiences with existing interfaces

For tasks where users were familiar with a similar interface, they would borrow elements from the existing solution. This led to nearly standardized sketches for the e-mail task. Thus, to foster exploration and creativity, novel tasks should be selected.

7.5. Colors are often used to show relationships and differences between objects.

We noticed a substantial usage of color in the first round by some users. While color had no standard meaning in most of the sketches, users attempted to differentiate object behavior and transitions based on color. From this observation, we felt that using color to link sketched elements would be an intuitive mapping for users.

8. Conclusion and Future Work

Our study and corresponding future work will have a noteworthy impact on next-generation sketching language design. Sketching tools and capabilities will expand, better modeling human expectations of the early design process. In addition to our prescriptive and comparative family of taxonomies, we have also developed a set of suggested notations, which could be used to extend existing tools.

Furthermore, we have identified color to be a linking mechanism between states, and will be exploring color-arrow hybrid notations as a solution to the “explosion of arrows” problem [LTL02]. Finally, we have identified an iterative investigation method, which facilitates refinement and extension of both the SNARF taxonomies and our suggested sketching notations.

User feedback validated the tasks we selected for our study and the resulting suggested notations. On average, participants using our notation set in the second round sketched about twice the amount of interface behavior when compared to the first round. The lessons we learned contribute towards understanding early design and pave the way for future investigators.

Extensions of our work promise to significantly increase the effectiveness of sketching techniques in early design. Additional iterations with large-scale user studies are required to further extend our taxonomies and identify additional notations. A diverse user base with regard to specialization and degree of expertise must be recruited in order to develop a more global outlook on the early design process. Additional novel and complex tasks must also be defined. The taxonomies and notations identified in this way may eventually be wrapped into sketching design patterns to provide common solutions for sketching language designers.

Follow-up studies may use a different procedure in order

to contrast their results with our suggested notations and taxonomies. For instance, using the SNARF taxonomies as a starting point, studies could require users to define their own concrete representation for each element in the taxonomies. A study such as this provides a more quantitative measure connecting the abstract desire with users' attempted representations. It may also identify and foster detailed investigation into points of variation such as color and arrow links.

References

- [BCR*03] BAUDISCH P., CUTRELL E., ROBBINS D., CZERWINSKI M., TANDLER P., BEDERSON B., ZIERLINGER A.: Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In *Proceedings of Interact 2003* (Sep. 1–5 2003).
- [BK03] BAILEY B. P., KONSTAN J. A.: Are informal tools better?: comparing demais, pencil and paper, and authorware for early multimedia design. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2003), ACM Press, pp. 313–320.
- [BKC01a] BAILEY B. P., KONSTAN J. A., CARLIS J. V.: Demais: designing multimedia applications with interactive storyboards. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia* (2001), ACM Press, pp. 241–250.
- [BKC01b] BAILEY B. P., KONSTAN J. A., CARLIS J. V.: Supporting multimedia designers: Towards more effective design tools. In *Multimedia Modeling* (2001), pp. 267–286.
- [CB04] COOK D. J., BAILEY B. P.: Understanding designers' use of paper in early design and its implication for informal tools, 2004. <http://orchid.cs.uiuc.edu/publications/tech-report-2004-2486.pdf>.
- [GD96] GROSS M. D., DO E. Y.-L.: Ambiguous intentions: a paper-like interface for creative design. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1996), ACM Press, pp. 183–192.
- [Hah04] HAHM H.-S.: Let nsync animate, 2004. <http://orchid.cs.uiuc.edu/projects/nsync/nsync-animate.wmv>.
- [Hea96] HEALEY C. G.: Choosing effective colours for data visualization. In *VIS '96: Proceedings of the 7th conference on Visualization '96* (Los Alamitos, CA, USA, 1996), IEEE Computer Society Press, pp. 263–ff.
- [Him98] HIMBERG J.: Enhancing the SOM-based data visualization by linking different data projections. In *Proceedings of 1st International Symposium IDEAL'98, Intelligent Data Engineering and Learning—Perspectives on Financial Engineering and Data Mining*. Springer, Hong Kong, 1998, pp. 427–434.
- [HK93] HERBSLEB J. D., KUWANA E.: Preserving knowledge in design projects: what designers need to know. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1993), ACM Press, pp. 7–14.
- [LM95] LANDAY J. A., MYERS B. A.: Interactive sketching for the early stages of user interface design. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), ACM Press/Addison-Wesley Publishing Co., pp. 43–50.
- [LNHL00] LIN J., NEWMAN M. W., HONG J. I., LANDAY J. A.: Denim: finding a tighter fit between tools and practice for web site design. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2000), ACM Press, pp. 510–517.
- [LTL02] LIN J., THOMSEN M., LANDAY J. A.: A visual language for sketching large and complex interactive designs. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems* (2002), ACM Press, pp. 307–314.
- [NL00] NEWMAN M. W., LANDAY J. A.: Sitemaps, storyboards, and specifications: a sketch of web site design practice. In *DIS '00: Proceedings of the conference on Designing interactive systems* (2000), ACM Press, pp. 263–274.
- [PA04] PLIMMER B., APPERLEY M.: Interacting with sketched interface designs: an evaluation study. In *CHI '04: Extended abstracts of the 2004 conference on Human factors and computing systems* (2004), ACM Press, pp. 1337–1340.
- [Rie99] RIEBER L. P.: Getting interactive with authorware: Building simulations and games, 1999. <http://www.nowhereroad.com/authorware/index.html>.
- [RSI96] RUDD J., STERN K., ISENSEE S.: Low vs. high-fidelity prototyping debate. *interactions* 3, 1 (1996), 76–85.
- [Ste] STEGER E.: Sketch-based animation language. <http://www.cs.toronto.edu/~esteger/sketchlang/sketch.pdf>.
- [vdL02] VAN DER LUGT R.: Functions of sketching in design idea generation meetings. In *C&C '02: Proceedings of the 4th conference on Creativity & cognition* (New York, NY, USA, 2002), ACM Press, pp. 72–79.
- [Wag90] WAGNER A.: Prototyping: A day in the life of an interface designer. In *The Art of Human-Computer Interface Design*, Laurel B., (Ed.). Addison-Wesley, 1990, pp. 79–84.

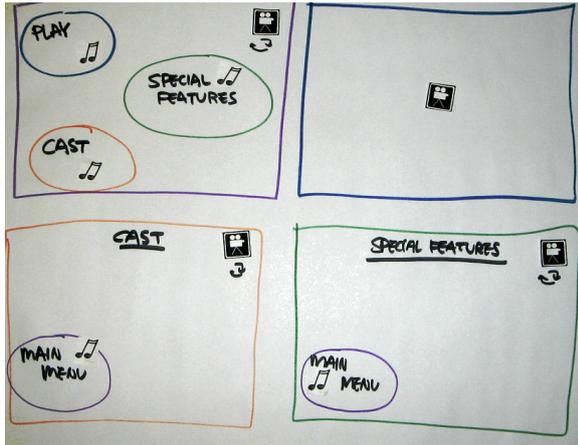


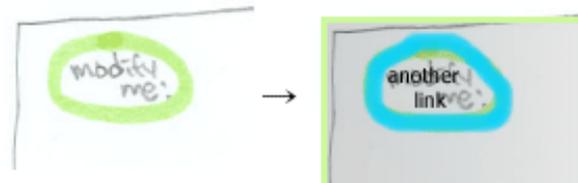
Figure 1: DVD interface designed using our sketching language.



Hover-Text/Mouseover Example: The text “mouseover text” will appear when the mouse cursor is placed over the red button.



Link Example: Clicking on the brown button labeled “link” transitions to the storyboard outlined in brown while clicking on the green button labeled “another link” opens the modal dialog outlined in green.



Sub-Overlay Change Example: Clicking on the green button labeled “modify me” opens the transparent overlay outlined in green. The button is now a light-blue button labeled “another link.”