How Designers Design and Prototype Interactive Behaviors

Brad Myers, Sun Young Park*, Yoko Nakano, Greg Mueller, Andrew Ko
*Human Computer Interaction Institute; *School of Design*
School of Computer Science
Carnegie Mellon University
bam@cs.cmu.edu; sunyoun1@andrew.cmu.edu; yokonakano@alumni.cmu.edu; gmueller@andrew.cmu.edu; ajko@cmu.edu

ABSTRACT
Designers are skilled at sketching and prototyping the look of interfaces, but to explore various behaviors (what the interface does) typically requires writing scripting code using Javascript, Flash or other programming tools. In our survey of 231 designers, 86% said that the behavior is more difficult to prototype than the appearance. Often (79% of the time), designing the behavior requires collaborating with developers, but 40% of designers reported that communicating the design to developers was difficult. Other results include that annotations such as arrows and paragraphs of text are used on top of sketches and storyboards to explain behaviors, and designers want to explore multiple versions of behaviors, but today's tools make this difficult. The results provide new ideas for future tools.

Author Keywords
Designer, Interactive Behaviors, Survey, Authoring, Flash.

ACM Classification Keywords
H5.2. Information Interfaces and Presentation – User Interfaces; D.2.6. Programming Environments.

INTRODUCTION
Designing user interfaces differs from designing static pages and movies in that user interfaces involve *interactivity*. Users click on buttons and links, fill in fields, directly manipulate graphical objects, and thereby control the results in a variety of ways. There have been many previous studies of the processes, techniques and tools that are used by designers, but none has focused on how the interactive behavior of the interface is created and communicated. Therefore, we conducted field studies of 13 designers, and a web-based survey, which received 231 responses, to investigate the particular issues for the design of interactive behaviors.

Submitted for publication

Do not cite or distribute

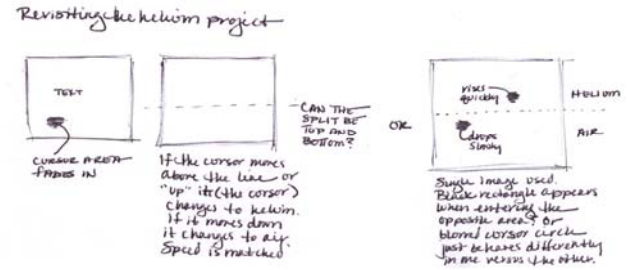

Figure 1: Sketches by a contextual inquiry participant showing two different options being investigated for an interaction, with lines and textual annotations to explain what is intended.

We particularly focused on people who are *not* programmers, but rather people who are trained and work on Interaction Design, Graphics Design, Information Architecture, Experience Design, Visual Design, User Interface Design, or equivalent.

In our survey, we defined an *interactive behavior* as:

> ...anything that an application does. If you think of "look and feel," we are mainly focusing on the "feel." Another way to look at it is that we are focusing on what you cannot draw in Photoshop, but instead need to describe in other ways, such as using storyboards, code, etc. Or, if you were developing in Flash or Director, the behavior would be anything that required using the timeline or scripting.

Many of our findings confirmed what others have reported in previous surveys (e.g., [3, 5, 13, 18]), for example that designers prefer to start by sketching (about 97% in our survey), and most designers (88% in our survey) also use storyboards.

However, we did find some interesting new results that have not been previously reported:

- By a large margin, the participants in our survey agreed that prototyping the behaviors was more difficult than the design of the appearance (86% said prototyping the behaviors was more difficult).
- Sketches and storyboards cannot adequately convey the behaviors by themselves, so designers must augment them with annotations such as arrows and many textual descriptions of the desired behaviors (see Figure 1).

- The purpose of implementing the interactive behaviors, and for the annotations on the pictures, is often primarily to serve as documentation and specifications for others. Almost all designers worked in teams, and communicating with others is a key part of their jobs. Communicating the design of behaviors to developers was reported to be difficult by 40% of the designers in our study.
- The behaviors that the designers wanted were quite complex and diverse, beyond what could plausibly be provided by a system that provided only a few built-in behaviors or a selection of predefined widgets, and therefore seemingly requires full programming capabilities.
- As reported for other kinds of design [3, 8, 17], in our survey, the designers agreed that the design of interactive behaviors emerge through the process of exploration. In other words, the designers do not have a final conception of the behavior before they start. However, whereas iterating on the look of the interface can be easily done by sketching, designers felt it difficult to iterate on the behavior. Today's authoring tools make it difficult or impossible to have two implementations of behaviors side-by-side to compare them, and even keeping around and reverting to old versions of code is difficult.

RELATED WORK

It is well known that designers prefer sketching for early phases of design [3, 13, 18]. Bill Buxton has devoted an entire book to this subject [3], which makes clear that sketching is an important technique for determining what should be designed, as well as for determining what the design should be. However, that book says little about determining the *behavior* of the designs, devoting only one page to interaction design. It gives some requirements for sketching behaviors, without any hint of how to achieve them:

> The tack that we are going to pursue is that sketching in interaction design can be thought of as analogous to traditional sketching. Since they need to be able to capture the essence of design concepts around transitions, dynamics, feel, phrasing, and all the other unique aspects of interactive systems, sketches of interaction must necessarily be distinct from the types of sketches we have looked at thus far. Nevertheless, to be considered sketches, they must be consistent with the attributes that we discussed earlier..." [3, p. 136]

A recent survey of 370 practitioners, both designers and developers, reported that "evolutionary prototyping" was the most common development process [4]. Tools used included paper and pencil, whiteboards, html editing, analysis and modeling tools, visual interface builders, etc. This survey reported that UI designers, like software developers, are constantly interrupted. The most difficult and costly transition reported was "moving from business rules, use cases, and problem space concepts into final solution design, and back," which is an area we did not explore in our survey.

Studies have looked at designers for particular domains, such as web authoring and animation. A study of 11 designers showed extensive use of informal tools in the early design phases for web sites [13]. None of these designers were involved in the programming for the final version of the web sites. To collaborate with developers, they used site maps, storyboards, annotations on top of sketches, and detailed Photoshop renderings. This study did not differentiate difficulties that designers had with the appearance from the behaviors.

A later study of 334 web developers who do not have professional training in programming showed that the collaborative group in which the developers worked was a greater contributor to their successes and failures than the tools they used [15]. Virtually all of the participants were self-taught for at least some of the programming skills they needed, and they often wanted to use behaviors that they could not successfully implement.

Other surveys have looked at designers of animation and multimedia. One study interviewed 7 animators and 8 non-animators [5]. The animators had extensive training, and about half used paper sketching and storyboards before switching to digital tools. The non-animators felt that creating animations would be too complicated. Another study interviewed 12 and surveyed 13 professional multimedia designers [1], and showed that a script was often the first thing created for multimedia, which is not an artifact mentioned for creating other types of design. The participants rated scripts, storyboards, sketching, and prototypes as important, but inadequate to relate important aspects of the dynamic aspects such as interactivity or timing.

Many people have created tools to help designers with sketching and authoring behaviors, often informed by the studies mentioned above. For example, tools for early design focused on use by designers who are not programmers include SILK [11], Electronic Cocktail Napkin [7], DENIM [13], SketchWizard [6], DEMAIS [1], Designer's Outpost [9], and many others. However, these tools mainly provide a limited fixed set of behaviors and are focused on particular domains.

METHOD

We employed the Contextual Inquiry (CI) methodology [2] with 13 participants. CIs involve observing and interviewing participants while they are in the process of working in their natural work environments to understand their practices and problems. For this study, we employed retrospective and artifact walkthroughs to investigate transitory work processes that would have been too time-consuming and impractical to study using the traditional CI method. In artifact and retrospective walkthrough, we asked participants to "walk through" their recent projects involving interactivity

using the resulting artifacts. Each CI took about 90 minutes and participants were not compensated.

We then validated the generalizabity of the CI data using a survey that was widely distributed. We sent requests to fill out the survey to our alumni mailing lists, to CHI-Announcements, CHI-RESOURCES, and the Interaction Design Association (IxDA) mailing list. This resulted in 231 responses, of which 181 (78%) finished all 47 questions. The complete survey took about 30 minutes to finish. The survey participants were entered into a raffle for five $25 gift certificates. To encourage people to answer the essay questions, we doubled the odds for people who filled them out completely, which worked pretty well – over half of the participants wrote comments in all of the fields.

RESULTS

We were pleased with the responses from our field study and survey participants. Overall, there seems to be a high interest in this topic, and people seemed very willing to help, and were hopeful there would eventually be interesting new tools as a result.

Demographics and Job Functions

Our participants had a wide variety of backgrounds and degrees.

The participants of the CIs were 4 university faculty, 1 Flash instructor, 3 Masters students, and 5 professional designers. Of the professionals, 4 worked as consultants and 1 as an in-house designer for a small company. The range of projects they have worked on recently included websites, web applications, desktop applications, mobile applications, multimedia, devices, research, and game design.

Most participants in the survey (about 45%) had a Masters degree, with 32% having their highest degree as a Bachelors. The area of the degree varied widely, and included psychology, human factors, anthropology, English, computer science, engineering, HCI, and many others. 17% had a CS degree, although 70% of them also had one or more degrees in the other areas mentioned (only 5% of the total participants *only* had CS degrees).

Their job titles at work included Interaction Designer (32%), User Interface Designer (20%), Information Architect (12%), and many others. About 6% were managers of groups that included designers. 9% had more than 20 years experience, 26% had 11-12 years experience, 36% had 6-10 years, 27% had 1-5 years, and 2% had less than one year.

Many (over 25%) worked in a design department of a company that consulted on various products or (22%) worked directly in a product division. About 21% worked for a design consultancy (a company that mostly consults with other companies), and about 13% worked alone. Many of the rest worked in universities and did design consultancy on the side. 60% worked for companies bigger than 100 people. About half spent most their time on websites, 6% on desktop applications, and 6% on phones.

Given that many worked in a consultancy role, we wondered if they got to see the products they worked on all the way through. In fact, 41% of the participants got to see the product that results from their designs "all the time," 34% saw it usually, 11% often, 12% sometimes, and only 2% saw it never.

Programming

We asked on the survey, "Is programming (also called 'scripting') a regular part of your job?" 61% answered "no." Interestingly, of the people who had a CS degree, 54% still answered "no," and for people without a CS degree, 64% answered "no." However, over 80% said they were good or expert with HTML, and 95% knew some other programming language at least to the level that they "can do a few things." About 34% said they were good or expert with Javascript, 16% with Flash, 15% with Java, 11% with Visual Basic, 11% with C++, 5% with C#, 8% with Lingo for Director, 18% with PHP, 3% with "Processing" (from www.processing.org), 1% with Ruby on Rails, and 1% with Cocoa & Objective C. The "Others" mentioned included CSS, Perl, XML, XSLT, Python, JSP, ASP, C, ColdFusion, LotusScript, VRML, Pascal, Max MSP, HyperCard, VoiceXML, and a few others.

We asked how they gained their programming experience, and 70% said they learned on the job to fulfill a need. 42% said they had taken some training classes in programming, 35% had pursued programming as a hobby outside of work, and 27% had a degree in programming. This is similar to the result of a previous study [15].

When participants in the survey had trouble understanding how to implement something, they usually "use Google or another search engine to search for examples" (17% sometimes, 28% often, and 46% usually do this). All of the other options we listed were also popular: "I go to the manual or books and look it up" was done sometimes 41%, often 21%, and usually 18%, "I go to on-line tutorials or on-line documentation" was used sometimes 24%, often 35%, and usually 32%, "I look for examples in code that I have around" was used sometimes 28%, often 39%, and usually 23%, and "I ask a colleague how to do it" was done sometimes 49%, often 21%, and usually 10%.

We asked *why* the participants had done programming on the survey, and got a range of answers, none of which dominated (see Table 1). This data seems to suggest a significant number of participants who do not prototype for any reason: 21% answered "rarely" or "never" to *all* choices.

Collaboration

In asking about the survey participant's "current or most recent project," 91% said they were working as part of a team. The role for most of the participants was as the user interface designer or the manager of the project. Other popular roles were information architect, artist, or web designer. The other members of the team typically included

other designers, a manager, developers, marketing people, business analysts, testers, usability analysts, and subject matter experts. Team size ranged from small (1-3) to huge (hundreds), with the median being about 5 people.

In the survey, participants reported that an important functionality they would look for in a new tool would include the ease of sharing and the ability to export designs into formats that can be easily viewed and commented on by others.

	Never	Rarely	Sometimes	Often	Frequently
To work out for myself a behavior I was not sure about	28%	23%	27%	12%	10%
To communicate a behavior to a developer who would implement it in the product	23%	15%	29%	20%	14%
To communicate a behavior to the clients or customers	20%	11%	24%	26%	20%
To use as part of a user test of the behaviors	22%	11%	22%	24%	21%

Table 1: How often participants in the survey created interactive prototypes for various purposes (N=184).

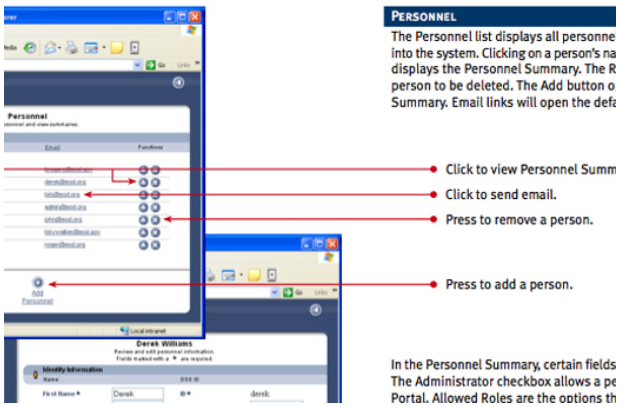

Figure 2: Annotations from CI participant U4 on top of a detailed design to serve as a specification for developers.

Communication with Developers

Given that most of the participants did not program, it is not surprising that 79% were working on a team that included a developer. Our CIs suggested that a primary job of the designer is to communicate and collaborate with the developer. One CI participant mentioned how much easier it was to collaborate with one developer than another—there were enormous individual differences. Most participants preferred to work face-to-face, but often their final deliverable would be a detailed design document to hand off to the developer. These documents would typically use many paragraphs of text and description of details around the pictures to explain the behaviors (see, for example, Figure 2). After delivering the documents, the designers would assume more of a support role, clarifying designs to developers as they implemented them. An in-house designer mentioned that he keeps a printed copy of the most recent version of each design in case a developer would come to him with questions. The CI participants emphasized that they considered the detailed design document to be an important deliverable, and they spent significant time making sure it was clear and that it looked polished.

These findings were confirmed in the survey. We asked about the frequency of collaboration with developers. The most popular answer was that the participant and developer "meet multiple times throughout the week" (39%), followed by "We work side-by-side" (18%), "We meet multiple times per day" (13%), "We meet once a week" (10%), "We meet infrequently" (7%), "We meet a few times a month" (5%), and "We only meet at the end of the project, when I delivered my results" (1%). The rest of the participants (7%) said this was not relevant or applicable.

One of the observations of the 2007 National Research Council report on human-system integration was that communication among various stakeholders needs to be improved [14]. In our survey, 40% agreed or strongly agreed with the statement: "Communicating the designs of the behaviors to developers is difficult," with 35% disagreeing, and the rest (25%) neither agreeing nor disagreeing. However, participants reported less difficulty communicating the appearance of the interface, with only 15% agreeing or strongly agreeing that this was difficult, and 51% disagreeing (with 25% neither agreeing nor disagreeing).

We investigated in the survey what techniques the participants used to communicate with developers. Although sketching is a key tool for designers, apparently it is *not* used as a communication tool: 56% of the participants *never* "draw sketches on paper then hand them off to the developer to build," with only 9% saying they did this "usually" or "all the time." In contrast, the most popular techniques used for communication were to "write textual descriptions of how the application will work" (78% do this at least "often"), followed by "I create static designs digitally (e.g., using Photoshop, or Web design tools like Dreamweaver or Frontpage) that I hand off to the developer to build" (66% do this at least "often"), "We collaborate around a whiteboard or sketchpad" (56%), "I create semi-functional, interactive prototypes (like in Flash) that I hand off to the developer to build" (33%), "I implement part of the final system by scripting or writing code" (22%), and "I create a movie" (4%).

Not surprisingly, when clarifications were needed by the developers of what the participants delivered, the most popular technique for clarifying what was meant was to discuss verbally (used "often," "usually" or "all the time" by 93%). Other popular techniques were to "Exchange emails or other textual explanations" (used at least "often" by 93%), "Edit the original deliverables, and deliver them again" (used at least "often" by 66%), and "Annotate on top of the original deliverables (e.g., sketch on a paper version of the original deliverable)" (42%). Less popular were to "Draw new sketches from scratch" (34%) and "Create a new deliverable from scratch" (11%).

For the statement, "I am often asked to help with detailed design questions about the appearance of the interface late

in the development process," 44% agreed and 61% disagreed, with 30% in the middle.

To get more details on this process, we provided an open response where we asked participants to describe situations where they had trouble communicating with developers, and what they did about it. Participants mentioned problems due to developers not being native English speakers, and the lack of details in the UI specifications which seem to be solved by face-to-face, phone, IM, and email conversations. One interesting comment was that "we publish our spec as a wiki – thereby allowing developers to add their annotations, questions and comments which we can in turn address. This makes deliverables more of 'living documents' than static paper weights."

However, in the end, participants felt that the result closely matched what they designed: 20% said the resulting product was "almost exactly as I specified," and 62% said it was "pretty close to my specifications," with only 13% saying it was "considerably different from my specifications" and nobody (0%) saying it was "nothing like my specifications" (the rest (5%) chose "not applicable / didn't check").

Differences between Behavior and Appearance

An important focus of our investigations is differences in the difficulty and techniques used for the appearance (look) versus the behavior (feel). Our CIs suggested that designers have more difficulty prototyping interactive behaviors compared to prototyping appearance. Prototyping of behaviors was reported to be an ongoing process while prototyping appearance is a simpler process of creating a single static image. People can imagine the final detailed appearance even from a vague sketch; however, it is very hard to communicate behavior and to get a sense of the final version from a prototype. Behaviors were said to often remain ill-defined until the application is actually implemented. Designers also mentioned frequently that because of the difficulties involved in communicating behavior, versions of the behavior are much harder to iterate on. By contrast, variations in appearance are easy to iterate on and can even be changed at the very end of the design process. Finally, some designers felt that a proper explanation of behavior can sometimes only be accomplished verbally. Interactive behaviors need many words to specify. Designers use scripts and text on each image to explain how it behaves; simply drawing a picture is never enough for them to explain the interactivity. Proper explanation requires words with multiple screens to show the states and the interaction after getting a user's feedback.

These results were supported by the survey, where 86% of the participants listing the behavior as more difficult to prototype than the appearance (see Figure 3). People with CS degrees answered "Behavior" 70% of the time, and people without CS degrees answered "Behavior" 84%. When we asked if the appearance of the interface hard to specify, nobody (0%) strongly agreed, and only 9% agreed, compared to 68% who disagreed (with 23% neither agreeing nor disagreeing). However, for the behavior, the results are reversed, with 44% agreeing and 38% disagreeing (with 18% neither).

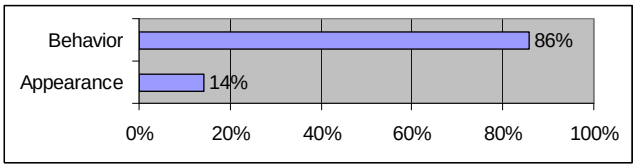

Figure 3: "Regarding Appearance versus Behavior, which is more difficult to prototype?"

To answer "why" the behaviors was more difficult, participants gave a variety of reasons. Here are some quotes:

- Lack of predefined behaviors and difficulty to introduce new ones.
- It starts as a blank slate.
- Because of the many different states that must be demonstrated and their dependence on different conditions preceding those states.
- Behaviors are often much more contextual and transient.
- Duh.
- We work on edge cases that appear rarely and in complex situations. Sometimes it involves things that are hard to prototype, like voice calls and camera shots.
- It's more complex with more variables and constraints attached. There are also more stakeholders and coworkers involved
- I find that I often want to prototype rich interactions, such as a sliding dock or an object that can be dragged and dropped. These are hard to illustrate in static documents.
- All the specific options must be communicated.
- It's harder for people to fill in the gaps with imagination. Small changes make big differences in experiential outcomes so if something is not quite right it can cloud whole thing.
- I work in mobile, so often I have to wait until developers have something working before I can actually test out how a new behavior 'feels' on a phone. What it looks like is easy to test by just putting a picture of the mockup onto the phone.
- Current tools for defining behavior suck.
- There's no such thing as low-fidelity interaction, it has to be right.
- Neither is difficult...just time consuming.

Similarly, 75% of the participants said that the behavior is more difficult to communicate to developers than the appearance. When we asked if communicating the designs of the appearance to developers is difficult, only 15% agreed and 61% disagreed. However, for the behavior, 40% agreed it was difficult to communicate, compared with 35% who said it was not.

On the question of when clarifications were required, there was an interesting difference between the responses for

appearance versus behavior: For the statement, "I am often asked to help with detailed design questions about the *behavior* of the interface late in the development process," 58% agreed and only 15% disagreed, with 27% in the middle. This is in contrast for the same question about the *appearance*, where the results were that 44% agreed and 26% disagreed. Participants agreed with the statement about behavior significantly more often than the one about appearance (chi square=65.9, $p < .0001$).

We asked participants how they divided their time between the behavior and the appearance, and the results ranged from 0% to 100% for each. Participants reported spending significantly more time on behavior ($t=6.8$, $p < .0001$). The two distributions for appearance and behavior were normally distributed. The mean and standard deviation for appearance are 39% (± 21), and for behavior are 61% (± 21).

Complexity of Behaviors

A goal of some prior research systems has been to make it easier to investigate interactive behaviors (e.g., [1, 6, 11, 13]), usually by simplifying the kinds of behaviors that can be expressed. We are skeptical of this approach because of the wide variety of behaviors we saw in our investigation. For example, in our CIs, the participants were creating interfaces that use 3D-rotation, physical simulations of dominos falling onto each other, interactions among graphical objects on the screen such as bouncing off each other, graphics that changed based on various sensors, novel physical devices, etc.

In the survey, we asked for ideas of what a future tool must support, and participants requested many things, including "character animation (using jointed inverse kinematics)," support for "more dynamic content, like AJAX," "database or CMS integration; for example, showing different content to different users based on business rules," "data-driven interactions," "ability to create RIA functionality" (Rich Internet Applications – web applications that act like traditional desktop applications), and "ease of ... connect[ing] to live data sources."

54% of the participants said that they had had an interactive behavior they wanted to explore on paper but could not, and therefore had to explore by implementing it. When asked to describe what they wanted to do but had difficulty with, the participants listed 96 behaviors. Here are some quotes:

- Camera interaction.
- On a recent project, there was a complex page with a lot of hide/show behavior. It was hard to get a sense of whether it would "work" by looking at static mockups -- a page with javascript made it much easier to see when the transitions were jarring.
- Synchronised behaviours.
- Issues around scrolling long pages. The experience of a long bit of paper isn't really the same for a user.
- A button that changes color to draw attention to itself.
- The use of window/panel transparency within the OS and various applications.
- Dynamic navigation systems (e.g. accordion menus), 3D or graphical navigational systems.
- Mobile interaction.
- More advanced interactions like drag-and-drop, column sorting, resizing on rollover, etc.
- Hover effect on a graphic, that when clicked, also selected a visible tab.
- Detailed keyboard-level interactions.
- Interactive art - engaging experiences.
- User experience of interaction with graphs.
- The exact timing of certain interactions.
- An animated 'lens effect' list UI.
- Gaming behaviour.
- Multi-dimensional selections that impact the display of other controls and data.
- Dynamic layout based on user preferences.
- Touchscreen displays.

However, these complexities do not seem to be preventing the participants from doing what they want. When asked if they agreed with the statement that "I change the design of the **behavior** to match whatever is easiest to implement," only 14% agreed and 63% disagreed with the rest (23%) neither agreeing nor disagreeing. This is similar to the responses for "I change the design of the **appearance** to match whatever is easiest to draw:" 8% agreed and 80% disagreed. The difference is statistically significant—more participants felt they had to change the behavior than the appearance (chi-square=180, $p < .0001$). We feel, however, that these numbers probably underestimate how much this happens. The faculty we interviewed in the CIs mentioned that people always adapt to their tools, sometimes unconsciously.

Phases and Process

Our CIs showed that participants use the "classic" phases for development (user research, sketching, prototyping, etc.), as also reported by prior studies [3, 5, 13, 18], but that there was a variety of names for the phases and ways they were practiced. Designers in the CIs unanimously focused on the content, structure and navigations before they explored the detailed visual appearance. One participant did not "want to distract the user with visual elements when working on the navigational structures" while still in the phase of receiving feedback and tweaking the structure. We observed that there interactive behaviors had to be explored throughout the design process--it must be included in early stages when navigational structures being worked out because interactivity is often involved in changing among states. However, the subtle details of interactive elements are tweaked until the end, even after the final graphics are implemented.

On the survey, we asked participants when creating a new product, what percent of the time would they personally want to spend on each of eight phases, and the average re-

sponses were: “Background research / reading” (9% ±7), “Competitive analysis” (6% ±5), “User Research / Contextual Inquiry” (14% ±10), “Ideation” (15% ±11), “Prototyping” (18% ±13), “Design Production / UI Development / Digital design production” (16% ±13), “Production code / Final Implementation” (9% ±11), “User Testing” (10% ±8). Participants were able to allocate 97% of their time to these phases, but also mentioned a few others that they spend time on, including: “documentation,” “spec writing,” “personas,” “client contact,” “project scoping and planning,” “use case analysis,” and “task design.”

The participants did propose a variety of different names for the phases, however. “Competitive analysis” was called “market research” and “work analysis.” “User research” was called “CTA (cognitive task analysis).” “Ideation” was called many things, including “wireframes,” “system design and/or information architecture,” “paper prototyping,” “low-fi prototyping,” “brainstorming,” “sketching,” “abstract design,” “concept development,” “conceptual design,” “concept work,” “conceptualization,” “conceptual modeling,” and even just “design.” “Prototyping” was called “mockups.” “User testing” was called “usability research.” Some participants mentioned specific models they followed, including the “ADDIE model (Analysis, Design, Development, Implementation and Evaluation)” [12], the “LUCID Framework” [10], and “OVID (Object, View, and Interaction Design) from IBM.”

We separately asked about what *techniques* the participants used, across all phases. Not surprisingly, “sketching” and “screen comps / mock ups” were the most commonly used (see Table 2). This matches results from previous surveys (e.g., [13]), which have reported use of sketches, site maps, storyboards (but less often than site maps), schematics, etc.

	Never	Sometimes	Often	Usually	All the time
Sketching	3%	18%	22%	21%	36%
Storyboarding	12%	30%	24%	20%	14%
Scenarios	8%	21%	27%	31%	13%
Timelines	36%	41%	15%	4%	4%
Wireframes	12%	11%	10%	20%	48%
Screen Comps / Mock ups	3%	8%	14%	27%	48%
Information Architecture Diagrams	14%	25%	19%	18%	24%
State Diagrams	31%	35%	17%	12%	5%
Flowcharts	12%	26%	26%	21%	15%
Sitemaps or screen transition diagrams	14%	26%	20%	20%	20%

Table 2: How often participants in the survey typically used each of these techniques when working on a project (N=187).

“Sketching” was used at least “often” by the participants of our survey 79% of the time, “screen comps” were used 89% of the time, “wireframes” 78%, “scenarios” 71%, “flowcharts” 62%, “information architecture diagrams” 61%, “storyboarding” 58%, “sitemaps or screen transition diagrams” 60%, “state diagrams” 44%, “timelines” 23%. It is interesting that “Timelines” were not commonly used, since this is the primary metaphor in the popular Adobe Director and Flash tools, suggesting that this technique might not match well with interaction design.

We asked if participants had different names for these techniques. One participant said “scenarios = use cases,” and another said “mock ups = wireframes.”

Other techniques mentioned by participants included: “documentation,” “written specifications,” “use cases and activity diagrams,” “affinity diagrams,” “UML models, HTA (hierarchical task analysis), Swim lane models, Design rationale (QOC),” “card sorting with post-it notes followed by word processor for functional design listings,” “wizard of oz and paper based prototyping in more advanced stages of the design,” “vision clips conversation scripting,” “animated walkthroughs,” and “comic strips.”

Tools Used

In our field study and survey, we saw that designers use a wide variety of tools, depending on the stage of the design process and their skills. For ideation sketches, all of our CI participants used paper and whiteboards. In designing navigational structures, Photoshop, Illustrator, and FreeHand were mentioned for creating wireframes. For visual design, a wide variety of tools were used including AfterEffect, PowerPoint, ImageReady, Dreamweaver, Flash and Director. In the more collaborative work settings, ease of use and ability to combine parts created by different people determined which tool to use. In one company, PowerPoint was used since everyone could share the results, although the designer complained that updating changes was a hassle, since they are not reflected in all of the files.

In the survey, the most popular tool mentioned was Photoshop, used at least “sometimes” by 86% in the survey, with 34% saying they use it almost every day. Other tools used included Dreamweaver (70% used it at least “sometimes”), PowerPoint (54% used it for mocking up interfaces), Illustrator (52% used it at least “sometimes”), Flash (43%), Visio (40%), InDesign (25%), Omnigraffle (21%), VisualStudio (17%), Fireworks (16%), Director (14%), Frontpage (12%), AfterEffects (9%), Axure (8%), GoLive (6%), Flex (4%), and Microsoft Expression Blend (3%).

Because the CI participants mentioned annoyance with the differences among all the tools they use, we asked about this on the survey. For the statement, “I find the differences in how different tools work to be frustrating,” 18% of the participants in the survey strongly agreed, 42% agreed (totaling 60%), with only 6% strongly disagreeing and 14% disagreeing (totaling 20%), with the rest (20%) neutral.

Transitions between Tools

Our CI participants indicated that knowing when to switch between different tools was a difficult decision. Most reported that they rely on their experience to decide.

We tried to find out how our survey participants decided when to switch. The most popular answer was “When I am happy with the result” (43%), followed by “When the pre-

determined deliverables are finished" (11%), "When the client is happy with the current phase" (10%), "When it passes user tests" (9%), and "When we've run out of time allocated to this phase" (8%). 19% said "Other," specifying various combinations of the above reasons, along with others including "when requirements are mature," "when the medium employed cannot further advance the idea," "when all the team members are in sync with the selected idea," and "when it's time to share results."

When asked "How much do you sketch on paper or on a whiteboard before going digital?" the most popular answer was "I sketch *a few* rough ideas on paper then move to digital," which was selected by 47% of the participants. 31% said "I spend *a lot* of time sketching out ideas on paper before going digital," 16% said "I *might* sketch on paper before going digital," and only 7% said "I prefer going straight to digital design."

Asked whether they "find it difficult to move artifacts and designs in between tools," participants in our survey were mixed: 43% agreed, 30% disagreed, and 27% were neutral. This is similar to the results of a previous study [4] which asked for ratings of frequency and difficulty of various transitions, and reported that the most frequent and least costly was moving from high-level descriptions to detailed screens.

Multiple Versions and Reuse

It is well-known that designers sketch multiple variations when exploring designs (see Figure 1). Buxton quotes Linus Pauling saying that "the best way to a good idea is to have lots of ideas" [3, p. 121]. Other studies have also documented this [13]. We were interested in how multiple versions were used for exploring interactive behaviors. In the CIs, one participant noted that "most ideas are bad and you want to get the bad ones out quickly" by iterating quickly. Designers said they rarely changed their designs drastically once the designs are in digital tools.

Surprisingly, in our survey, participants did not indicate extensive exploration of different designs. We asked "How many different ideas do you typically explore in early design," with 60% choosing "I explore only a few designs." 37% chose "I try to explore many different designs before choosing which to concentrate on" and 2% chose "I usually know the final design from the start." This matches the answer mentioned above that only 31% said they "spend a lot of time sketching out ideas on paper before going digital."

We asked participants how many different versions of the appearance and behaviors they explored in their current or most recent project (see Table 3). Although the most popular answer was "between 2-5" for both appearance and behaviors, participants reported exploring more versions for behavior than appearance ($t=3.2$, $p < .01$). We feel that this number is surprisingly low. Possibly, the participants were interpreting the word "version" in the question too strongly: our CI participants discussed many iterations on aspects of the appearance: changing colors, placements, etc. that survey participants may not have considered new "versions." This may have decreased the counts for appearance, whereas our CIs show that the iterations for behaviors are often caused by negotiations with developers, and may make different "versions" be more salient.

	1	**2-5**	**6-10**	**11-30**	**31-50**	**>50**
Appearance	10%	71%	11%	6%	1%	1%
Behavior	8%	59%	22%	10%	1%	1%

Table 3: How many different versions of the appearance and behavior the participants explored (N=210).

As shown in Figure 1, we saw in our CIs that designers frequently wanted to have multiple designs side-by-side, either in their sketch books or on big displays or on the wall. However, this is difficult to achieve for behaviors – there is no built-in way to have two versions of a behavior operating side-by-side. Therefore we asked in the survey, "How important is it to you to be able to compare alternatives to decide which one to use?" 53% of the participants rated this "very important" or "crucial" for sketches, compared to 57% for wireframes, 54% for detailed images, 48% for layouts / grids, 43% for implemented user interfaces, 27% for storyboards, and 12% for code examples.

Programmers rarely start a new project from scratch, but instead reuse old code as a starting point. We wondered if this applied to designers as well. In the CIs, we saw one participant having a library of templates in tools such as Photoshop and Illustrator to save time creating standard widgets. A previous study of web developers [15] showed that non-programmers were less likely to reuse their own code.

In the survey, we asked, "For the following kinds of design elements and artifacts that you created for a project, how often do you reuse them for a later project?" As expected, sketches are almost never reused – only 5% said "often," 4% said "usually," and 1% said "all the time." Similarly, storyboards are reused at least "often" only 12% of the time. The answer for sounds is 14%, wireframes 34%, detailed graphic elements 38%, images / photos 39%, code / scripts 46% (which confirms the previous result [15]), colors / palettes 51%, layouts / grids 52%, icons 58%, templates / style sheets 62%, and Widgets / Controls / Interaction Techniques (e.g., menus, buttons, sliders) 76%.

We noted in the CIs that users did not have any good way of saving and reusing different versions of their code, either to enable comparisons in the current project, or to save for future projects. They had to use various naming schemes and file structures of their own invention. This was evident when a participant had a hard time locating the file he wanted to show the researchers during a CI.

Therefore, we wanted to see in the survey what participants did to keep track of previous versions of the code for a project. Choices included: "I name the old versions with different file names" which was used "never" by 25%, "some-

times" by 28%, "often" by 19% and "usually" by 28%; "I use different folders for different versions," used "never" by 32%, "sometimes" by 29%, "often" by 20% and "usually" by 19%; "I copy old versions to external media, like CDs or a file server" which was used "never" by 55%, "sometimes" by 25%, "often" by 8% and "usually" by 13%; and "I use an electronic version tracking / version control system such as CVS or Visual SourceSafe" which was used "never" by 51%, "sometimes" by 20%, "often" by 9% and "usually" by 19%. Of all the participants, 88% were using at least one of these techniques at least some time.

Annotation and Documentation

Previous studies have reported that storyboards, by themselves, are usually not sufficient for describing interactive behaviors and often require additional information such as gestures, face-to-face conversation, and graphical and textual descriptions [3, p. 180, etc.]. Bailey [1] describes multimedia designers' use of annotations on storyboards to provide a rich source of behavior definition. We observed during our CIs that designers use annotation extensively on sketches, storyboards, wireframes, and formal design documents to communicate their ideas more clearly to their colleagues, to developers, and to other stakeholders (see Figure 1 and Figure 2). The kinds of annotations we observed included labels, arrows, and narrative textual descriptions. Designers use more arrows at sketching and wireframe stages and also in flow charts to indicate navigational transitions between widgets and controls.

Annotated design documents serve both as important deliverables to clients and a way to record the history of design decisions for the designers to review later. CIs revealed that designers spend considerable amount of time creating, formatting and updating these documents. We were told that clients' demands for frequent updates to the documents to reflect the ongoing changes interfered with designers' creative workflow. Some designers expressed a desire to automate keeping documents in sync with changes to other artifacts, but they were not willing to give up any control over the appearance and structure of the documents since they regard these documents as an important part of their work.

Our survey confirmed that annotations are important to designers. We asked, "After completing a sketch or digital drawing, how often do you use the following techniques to help explain to others how it behaves." For "Drawing arrows and connecting lines on top of and between the pictures," only 12% said "never," with 59% using this "often," "usually" or "all the time" (the rest (29%) saying "sometimes"). For "Adding annotations and small pieces of text as explanation on a picture," only 3% never used this, but 82% use it at least "often." For "Writing paragraphs of text about a picture," 18% never do this, but 52% do it at least "often." For "Putting the pictures into a presentation (like PowerPoint)," 16% never used this, but 52% use it at least "often." For "Making a movie or animation with the pictures to explain the behaviors," 52% never do this, but 23% do it at least "often." For "Creating click through mockups or interactive prototypes," only 9% never do this, but 64% use it at least "often." In the comments on this question, one participant emphasized: "[I] SCAN my drawings to PDF and combine with design notes: I NEED to TYPE and SKETCH on the same paper." Another noted:

> I can put all the text in that I want but only 5% of the people actually read them [but they] read every detail. They are mostly notes to me for later documentation purposes or when I explain them to people or to explain alternatives ... I can show that I thought of alternatives to the current design (they will think I am incompetent if they think of something I don't).

In the survey, participants also expressed the need for content management capability and ease of documenting, updating and version control for tracking design changes, sharing them with developers and clients, and user testing of the documentation. One participant wished for a tool that would take an "object-oriented approach to wireframing, ... where changes to interface elements would automatically be reflected wherever they appear." Another said, "Word is probably our most important tool because the text support allows us to easily describe UI behavior in great detail and everyone can open the file and edit it if needed... However, formatting in Word is a huge pain."

DISCUSSION

Our results suggest that there is definitely a need among designers for quickly and easily prototyping interactive behaviors. Many do not program, and have difficulty prototyping behaviors. Consequently, they resort to heavy use of annotations and other means for communicating their intentions. Specifying behaviors textually is often inadequate, time-consuming and sometimes leads to miscommunication and confusion between designers and developers. Since designers are always striving to be creative and explore new and complex behaviors, this is a significant pain point for designers both today and in the future.

Threats to Validity

There are a number of reasons our results may not generalize. In our CIs and survey, we tried to focus on interaction designers, but we cannot be sure whether the participants are representative of the entire field, or who filled out our survey. Our means of reaching designers may have biased the results towards the views taught at our university (since we targeted our alumni), or who are members of a particular organization (ACM SIGCHI) and mailing list (IxDA). However, since our results match those previously reported, and since we seemed to have quite a wide variety of degrees and jobs represented, we feel comfortable relying on the results.

Another possible criticism is that the discussion about communication and annotations should include the perspective of what the *receivers* want from designers, so it might be wise to also survey developers. It is also important to

note that some developers do not collaborate with designers.

Implications for Future Tools

The results reported here provide clear requirements for new tools, which go beyond what any of today's commercial or research tools offer. Only 1 person out of the 244 we interviewed or surveyed said he is *not* interested in a new tool.

Designers are very comfortable sketching and using digital tools for the look of the interface, but still are severely lacking in appropriate tools for exploring behaviors. They would like to iterate and explore multiple behaviors themselves, but are inhibited by today's tools. However, just providing a few behaviors in menus for the designers to pick from is not sufficient, since designers want to explore quite complex behaviors.

Future tools should also provide support for exploring different versions of the behaviors. Today's tools make the user be responsible for naming multiple versions of files, and often do not even let designers compare multiple versions side-by-side. Terry provides an example of how this might be done [16].

There are significant opportunities for supporting tasks for the behaviors beyond programming them. No tool today makes it easy to annotate and describe the behaviors, which is important for facilitating communication about the behaviors with developers. And since many designers work in teams with other designers, tools to facilitate collaboration around the exploration might be helpful. Many designers expressed the need for their tools to better integrate with other kinds of tools, like with Visio and Photoshop, to avoid redundant work.

CONCLUSIONS

Our contextual inquiries and surveys have confirmed what others have reported, but also revealed new requirements and ideas. Although time-consuming, doing these kinds of user research before embarking on new tool efforts is very important for ensuring that the results will actually be helpful for the target audience. We hope the information presented here will enable us and others to produce such tools.

ACKNOWLEDGMENTS

We thank Jodi Forlizzi, John Zimmerman, and all the anonymous participants for help with this work. Thanks to Adobe for financial support and advice for this research.

REFERENCES

[1] Bailey, B.P., Konstan, J.A., and Carlis, J.V. "Supporting Multimedia Designers: Towards more Effective Design Tools," in *8th International Conference on Multimedia Modeling.* 2001. pp. 267-286.

[2] Beyer, H. and Holtzblatt, K., *Contextual Design: Defining Custom-Centered Systems.* 1998, San Francisco, CA: Morgan Kaufmann Publishers, Inc.

[3] Buxton, B., *Sketching User Experiences: Getting the Design Right and the Right Design.* 2007, San Francisco, CA: Morgan Kaufmann.

[4] Campos, P. and Nunes, N.J., "Practitioner Tools and Workstyles for User-Interface Design." *IEEE Software*, January/February, 2007. **24**(1): pp. 73-80.

[5] Davis, R.C. and Landay, J.A. "Informal Animation Sketching: Requirements and Design," in *AAAI 2004 Fall Symposium on Making Pen-Based Interaction Intelligent and Natural.* October 21-24, 2004. pp. 42-48.

[6] Davis, R.C., Saponas, T.S., Shilman, M., and Landay, J.A. "SketchWizard: Wizard of Oz Prototyping of Pen-based User Interfaces," in *UIST 2007.* October 7-10, 2007. Newport, RI: pp. To appear.

[7] Gross, M.D. and Do, E.Y.L. "Demonstrating the electronic cocktail napkin: a paper-like interface for early design," in *Conference companion for CHI'96: Human factors in computing systems.* 1996. Vancouver, British Columbia, Canada: pp. 5-6.

[8] Henderson, K., *On Line and On Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering.* 1999, Cambridge, MA: MIT Press.

[9] Klemmer, S.R., Thomsen, M., Phelps-Goodman, E., Lee, R., and Landay, J.A. "Where do web sites come from? capturing and interacting with design history," in *Proceedings of CHI'2002: the SIGCHI conference on Human factors in computing systems.* 2002. Minneapolis, Minnesota: pp. 1-8.

[10] Kreitzberg, C.B., "The LUCID Framework™," 2007. Cognetics Corporation: http://www.cognetics.com/lucid/.

[11] Landay, J. and Myers, B., "Sketching Interfaces: Toward More Human Interface Design." *IEEE Computer*, March, 2001. **34**(3): pp. 56-64.

[12] Molenda, M. "In Search of the Elusive ADDIE Model." *Performance improvement*, May/June, 2003. **42**(5): pp. 34-36.

[13] Newman, M.W. and Landay, J.A. "Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice," in *Designing Interactive Systems, DIS 2000.* August, 2000. New York City: pp. 263-274.

[14] Pew, R.W. and Mavor, A.S., eds. *Human-System Integration in the System Development Process: A New Look.* 2007, The National Academies Press: Washington, D.C. National Research Council.

[15] Rosson, M.B., Ballin, J., and Rode, J. "Who, What, and How: A Survey of Informal and Professional Web Developers," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05).* 2005. pp. 199 - 206.

[16] Terry, M. and Mynatt, E. "Recognizing creative needs in user interface design," in *C&C'02: Proceedings of the ACM Conference on Creativity and Cognition.* 2002. Loughborough, United Kingdom: pp. 38-44.

[17] Tversky, B., Suwa, M., Agrawala, M., H. J, C.S., Hanrahan, P., Phan, D., Klingner, J., Daniel, M., Lee, P., and Haymaker, J., "Human behavior in design: Individuals, teams, tools," in *Sketches for Design and Design of Sketches*, 2003, Springer. pp. 79–86.

[18] Wong, Y.Y. "Rough and Ready Prototypes: Lessons from Graphic Design," in *Extended Abstracts, SIGCHI'92.* May, 1992. Monterey, CA: pp. 685.