Entangled Design Knowledge: Relationships as an Approach to Claims Reuse

Shahtab Wahid, D. Scott McCrickard, C. M. Chewar, Jason Chong Lee

Virginia Polytechnic Institute and State University, United States Military Academy

Center for Human-Computer Interaction
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0106
{swahid, mccricks, chonglee}@cs.vt.edu

Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, NY 10996
christa.chewar@usma.edu

Shahtab Wahid is a computer scientist with an interest in HCI design knowledge reuse; he is a Ph.D. candidate in the Department of Computer Science at Virginia Tech, Blacksburg, VA, USA. D. Scott McCrickard is a computer scientist with interests in notification systems, methods in HCI design, and reuse; he is an Associate Professor in the Department of Computer Science and is a member of the Center for HCI at Virginia Tech, Blacksburg, VA, USA. C. M. Chewar is a computer scientist with an interest in notification systems, methods in HCI design, and reuse; she is an Academy Professor in the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point, NY, USA. Jason Chong Lee is a computer scientist with interests in usability and agile software development methodologies and design rationale; he is a Ph.D. candidate in the Department of Computer Science at Virginia Tech, Blacksburg, VA, USA.
ABSTRACT

As a discipline, human-computer interaction produces innovative designs that could provide a reusable collection of design knowledge on which future efforts could build on. It is unfortunate that so much of this knowledge is not fully reused by designers today. To encourage the use of previously identified HCI knowledge, we propose a model of reuse building on Carroll’s notion of claims, design knowledge components that capture the positive and negative psychological effects of design features. We address four challenges associated with reuse adopted from software engineering—a discipline in which the notion of reuse has been prevalent for quite some time. Building on Krueger’s definition of reuse and his conceptualization of four key aspects—abstraction, selection, specification, and integration—we propose a reuse approach based on incorporating these four aspects into the design process. To abstract, select, specify and integrate claims, we identify claim relationships, descriptions of connections between claims. We portray how claim relationships can be used to aid in identifying claim types, searching for claims, creating new claims, and aggregating claims. We present three exploratory studies in which we explore how claim relationships can be used to find claims to reuse, create new claims, and illustrate the design of a system as a whole. We conclude with remarks on our work, discuss possible problems, and directions to take.
CONTENTS

1. INTRODUCTION

2. BACKGROUND AND MOTIVATION
   2.1. Reuse
   2.2. Design Knowledge: Claims
   2.3. Using Relationships

3. IMPETUS FOR REUSE IN HCI
   3.1. Requirement 1: Abstraction for Identification of Component Types
   3.2. Requirement 2: Selection of Library Contents
   3.3. Requirement 3: Specification to Adapt Components
   3.4. Requirement 4: Integration of Reused Components

4. CLAIM RELATIONSHIPS
   4.1. Postulating/Predicating Claims
   4.2. Executing/Evaluating Claims
   4.3. Generalizing/Specifying Claims
   4.4. Translating Claims
   4.5. Fusing/Diffusing Claims
   4.6. Mitigating Claims

5. REUSE ASPECTS AND CLAIM RELATIONSHIPS
   5.1. Claims Library
   5.2. Abstraction
   5.3. Selection
   5.4. Specification
   5.5. Integration
   5.6. Practicality

6. EXPLORING THE UTILITY OF CLAIM RELATIONSHIPS
   6.1. Exploring Abstraction and Selection
   6.2. Exploring Claim Selection
   6.3. Claims Map Creation

7. CONCLUSIONS
1. INTRODUCTION

Good practitioners of human-computer interaction (HCI) generally produce creative and innovative designs, but often ignored are incremental design approaches that combine and build upon established techniques and approaches. Our research approach seeks ways to capture the collective design knowledge of the HCI community and appropriately deliver pieces of this knowledge to designers. For example, an important finding, piece of knowledge, can be encapsulated in a form and then shown to designers so that they can account for the finding in their own work. It may potentially be easier to reuse this knowledge than create new design knowledge. Although this shows promise, we must strive to understand how one can both facilitate reuse and contribute reusable knowledge.

Systems based upon design knowledge previously identified and erected using fundamental theories and observational studies show promise in achieving higher quality designs. Instead of asking designers to create their own design from scratch and yield radical designs each time, we would like to see designers use previously identified design knowledge during their design process, giving them the ability to consider previous research and design efforts. Whittaker, Terveen, and Nardi (2000) argue radical invention is vital to making progress, but that designers should try to always make improvements based on prior work. It is argued that only when designers refer to prior work and can no longer improve upon it does it make more sense to consult radical invention. For this reason, there is a developing need to reuse design knowledge during the development process. Such a development method can potentially ease and decrease overheads such as time and cost in design. The benefits of reuse for HCI design knowledge were previously explored (Sutcliffe, 2000)(Sutcliffe & Carroll, 1999).

What should we be reusing? How can we enable reuse during design? How does one find the most appropriate design knowledge? Can we create components others can reuse? Is there a way to combine everything into a single design? These are pressing questions we aim to investigate in this work. Our goal is to use a design knowledge form created within the HCI community, claims, as the unit of reuse within a reuse methodology. Although we acknowledge there may be significant overhead to creating and maintaining claims, we believe there is great value in investigating reuse processes based on claims.

Much research on reuse was done within the software engineering community (Dusink and van Katwijk, 1995). To build our reuse methodology, we turn to Krueger’s (1992) notion of how to enable reuse based on four key concepts: abstraction, selection, specification, and integration. We draw from this methodology and present how we can use the same concepts for the reuse of claims. Central to the workings of our methodology is the idea of claim relationships, connections between claims.

We elaborate on our work in Section 2 by first discussing reuse, claims, and the use of relationships. Section 3 establishes Krueger’s vision of an approach to reuse, defines the four reuse aspects, and maps the aspects to requirements we needed to satisfy with
our methodology. We present definitions and examples for the claim relationships we base our method on in Section 4. Section 5 illustrates how our relationships can be applied in four different ways to exhibit characteristics of Krueger’s reuse aspects and provide potential solutions to the identified problems. Section 6 describes a series of exploratory studies to investigate the methodology. Finally, we provide closing thoughts in Section 6.

2. BACKGROUND

In this section we cover the topics upon which this research is based. We first explain the reasons to reuse and what is needed for it. This is used as motivation for the reuse of claims, the design knowledge components concerning our work. Finally, an overview of the concept of relationships and their usefulness in linking claims and aiding reuse is discussed.

2.1. Reuse

Reuse, the idea of activities reusing previously created artifacts consisting of pieces of formalized components, has attracted a lot of research. Much of this research was conducted within software engineering. Biggerstaff and Perlis (1989a & 1989b) compiled a collection representing early advances in models and applications of reuse. Dusink and van Katwijk (1995) maintain, in a broad survey of the research topic, the general claim regarding reuse, if applied properly, is that it can speed up the development process by reducing the amount of time and effort put into previously solved issues. However, tackling reuse as a research agenda has proven to be difficult in many ways. Gall and colleagues (1995) stated that after years of research the community was still not able to establish reuse as an integral part of development and that there was no agreement regarding the basic unit of reusability.

One unit of reusability that has been explored within software engineering is the pattern. Originally proposed by Alexander et al. (1977) for the design of buildings and towns, patterns are reusable components encapsulating knowledge. They include information such as context of use, conflicting forces, and potential solutions. Patterns were adopted by the software engineers and developed a following over the years largely due to Gamma, Helm, Johnson, and Vlissides (1995), the Gang of Four, who proposed the reuse of patterns for software development. These can now be used to describe software development problems and analysis tasks.

More recently a distinction was made between the traditional view of code reuse and an emerging trend (Ravichandran & Rothenberger, 2003). White-box reuse involves searching for code components, modifying them for use, and then depositing the component for others for use. This type of reuse involves the use of reuse repositories—an area for which many strategies and searching methods are researched (Henninger, 1997). Krueger (1992) proposed that such repositories must be based on the notion of abstraction, selection, specification, and integration (see Section 3). Black-box reuse, on the other hand, is a new trend that involves using components without modifications. Although black-box reuse requires that developers know the functionality of the
component and how to interface with it—typically for component-based development (CBD)—it reduces the need to search for and modify components.

Research on reuse in HCI is also done, but the topic does not have the large following the software engineering community has to date. There are a number of people that looked at the reuse of patterns for HCI. Borchers (2000) argued for the use of design patterns to capture HCI knowledge. He mentioned the need for the encapsulation of the designers’ experiences, methods, and values into patterns. Some have created patterns for specific domains. For example, Landay and Borriello (2003) created patterns for ubiquitous computing. Their goal was to apply them within a field by documenting lessons learned and passing them on to new designs.

Such research efforts into the reuse of patterns provide impetus behind the argument to consider other forms of reusable design knowledge to benefit HCI. Claims are a form of recording knowledge initially proposed by Carroll and Kellogg (1989) (see Section 2.2). Claims were even included as a part of a design pattern structure by Hughes (2007). Sutcliffe proposed and spearheaded efforts to make claims into reusable design knowledge components, lasting well beyond the designs they were initially created for. An extensive structure for claims and the idea of storing claims in a library was presented (Sutcliffe & Carroll, 1999)(Sutcliffe, 2000). This work is closer to the notion of white-box reuse. This is most likely because we have not found any evidence of the field having an established methodology such as CBD, allowing a form of black-box reuse to grow within HCI.

### 2.2. Design Knowledge: Claims

Scenario-Based Design (SBD) (Rosson & Carroll, 2002) is a design process in which scenarios, narratives describing a particular task, are used as a base for creating interactive designs. SBD uses various types of scenarios to guide the design process. An analytic evaluation process within SBD, called claims analysis, identifies scenario features that have usability consequences and stores this information in a structure called a claim. Carroll’s claims, used in the Task-Artifact Theory (Carroll, Kellogg, & Rosson, 1991), are design knowledge components which capture the positive and negative effects of an artifact within a usage context (Carroll, 1994)(Carroll and Kellogg, 1989)(Carroll, Singley, & Rosson, 1992). Delivered in informal natural language, claims address a variety of situational and interface aspects that affect the compatibility of the design and user models, such as user satisfaction and feeling of reward, color and object layout, and strength of affordances. Claims aim to provide designers with a pure view into what makes an artifact live and breathe, grounded in theories and observations of user experiences.

Carroll introduced claims as a way to capture design knowledge. Generally, the claim concept has been used as a disposable knowledge unit to guide conceptualization of a design. However, creating claims to record such insight introduces overhead to the design process. To demonstrate the model of a claim, we present one about the collage metaphor and assess its effects (see Figure 1). A collage metaphor stems from the notion
that artifacts are placed haphazardly in an unorganized fashion, much like a public bulletin board (Greenberg & Rounding, 2001).

<table>
<thead>
<tr>
<th>Organizing information items using a collage metaphor</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Allows users to informally post information without any regard to organization</td>
</tr>
<tr>
<td>+ Allows users to gain an understanding of an item's age/applicability with respect to the number of items that may be covering it</td>
</tr>
<tr>
<td>+ Lack of information categorization accommodates a wide range of different types of information to be placed</td>
</tr>
<tr>
<td>- BUT the lack of organization can hinder efforts to find a particular information item</td>
</tr>
<tr>
<td>- BUT overlapping items may force users to move items in order to fully reveal themselves</td>
</tr>
</tbody>
</table>

Figure 1. An example of a claim showing a title, upsides, and downsides.

The title or feature at the top of the claim summarizes what the claim is about or intended for. The following tradeoffs describe the consequences of the feature. One can imagine a situation where a designer is trying to apply an organizational technique to a public display. The construct of the claim in Figure 1 would allow a designer to assess the tradeoffs of using a collage metaphor in the design of the system. Along with other claims about organizational methods, the designer can choose the best option. Through this structure, the ideas portrayed within the claim can be passed along and reused by other designers, making claims reusable design knowledge components.

A developer can make use of this claim in various ways. First, the designer can gain insight into what the collage metaphor is. Since claims should be grounded in theories and observational studies, the designer can gauge the effects of using the feature in their own design. It is conceivable that one could think of an idea similar to the collage metaphor (without knowing what a collage metaphor is), but it may be hard to readily assess the effects of the design feature. With the claim tradeoffs (upsides and downsides) taken into account, not only do designers gain inspiration, but also get the opportunity to consider whether integrating such a feature into their own design would be appropriate. During the evaluation phases of their work, claim tradeoffs are indications of aspects of their design that should be tested. Second, designers creating claims engage in the process of identifying specific design aspects that can be captured in terms of a claim. They are forced to consider theories and conduct their own studies to assess the effects of a feature, enabling designers to base decisions on rationale instead of instinct.

2.3. Using Relationships

In his seminal book, Alexander et al. (1977) present patterns from the architecture domain ranging from higher-level city concerns, such as the distribution of towns, to lower-level issues, such as individual rooms in homes. Each pattern identifies problems regarding these issues and presents a solution. The solution itself can be comprised of other patterns. Hence, patterns are linked together to point to other patterns. Following
these links within the book can lead readers to other patterns that they may need, creating a method for browsing through the book. Although Alexander does not use relationship types, his notion is similar to the core of this paper.

Zimmer (1995) builds on Alexander’s work by attempting to apply relationship types to design patterns. He understood the notion that patterns were being linked to and even combined with each other without defining the nature of the association. He subsequently categorizes his relationships into three types: a pattern using another pattern, a pattern similar to another pattern, and a pattern combined with another pattern.

Digital libraries that exploit relationships to structure information are nothing new. Embly (1987) demonstrates a library containing abstract data types (ADT) for reuse that is structured to use automatic and user-defined relationships among entities. The objective is to locate ADTs in the library and use them in software that is under development. The structure of the relationships facilitate locating using automated functionality, browsing by following links, and software building related activities through integration, providing increased flexibility.

Creech, Freeze, and Griss (1991) focus on how to structure and efficiently select components in a reuse library. They consequently explore the use of hypertext in selecting reuse components. Their belief is that the appropriate use of hypertext to structure components would aid the selection process. By providing a graphical view of a library, users were allowed to browse through the library by navigating from component to component.

The most notable mention of relationships similar to our work can be found within the research done on claims. In the past, there have been assertions that for claims to be an effective component of reuse, they need to be classified and organized in a library. Previous work mentioned that there is a need for claim relationships within such a library to help bring structure and organization (Sutcliffe & Carroll, 1999). It is argued that relating claims can expose different levels of granularity and associating claims together can lead to the creation of new artifacts expressed as child claims (Sutcliffe, 2000).

Throughout these examples of uses of relationships we see how the need for linking pieces of design information together and understanding the nature of the link arises. They show how relationships are crucial to creating effective and usable libraries of reusable information. Through the application of relationships we can begin to see their benefits for developers and design.

3. IMPETUS FOR REUSE IN HCI

To instantiate a reuse solution within HCI we find it essential to analyze a strong solution from software engineering. Krueger (1992) successfully argues his own perspective on reuse. Many use this work as the de facto definition of reuse and its characteristics (e.g., (Ye & Fisher, 2002)(Basili, Briand, & Melo, 1996)(Johnson, 1997)(Sugumaran, Tanniru, & Storey, 2000)). Krueger suggests any approach to reuse must, “provide natural, succinct, high-level abstractions that describe artifacts in terms of
‘what’ they do rather than ‘how’ they do it (Krueger, 1992).” It is more important to first describe components in terms of what they mean to an overall design (such as a solution or alternative) instead of what the specific contribution is (a new layout method). For example, it is argued that it may be more beneficial if a designer knows a certain component is a potential solution to a design problem rather than knowing what the solution described by the component is. Our conceptualization of claim relationships, the characteristics describing interactions between claims, forms the basis of our approach to describe ‘what’ components do—a topic covered in Section 4.

Krueger’s broad analysis of reuse techniques demonstrates how each technique is based on four aspects crucial to facilitating reuse: abstraction, selection, specification, and integration. Abstraction is an essential feature to any reuse technique. The essence of abstraction is a succinct description concealing the unimportant details and emphasizing only information that is important. Without it, designers would be forced to meticulously peruse through assemblies of reusable components, prolonging the time spent looking for an appropriate component. Selection involves locating, choosing, and viewing an artifact for potential reuse. Classification or categorization methods may be used to organize a reuse library for artifacts to be found and reused by designers. In many cases, general components may need to be adapted or further specified, accentuating the need for specification. This is often an important step because designers may not always find exactly what they need and will be forced to adapt components. Finally, components must be gathered and integrated into a coherent design. Designers must understand how components that have not been used together before can interact with each other within the context of the overall design. Section 5 introduces how our concept of design knowledge relationships can be applied in terms of these four reuse aspects.

We establish the need for the reuse aspects by asserting they map to problems in facilitating claims reuse. Understanding these problems and associating them to all of the reuse aspects allows us to demonstrate a software engineering approach has the potential to help HCI knowledge reuse. To the best of our knowledge, we have not seen any other example of Krueger’s vision being brought into HCI research. We continue this section by presenting four requirements and demonstrate how they can be solved with the incorporation of the reuse aspects.

3.1. Requirement 1: Abstraction for Identification of Component Types

While the structure of a claim is very simple, claims can be written and used in a variety of different ways, leading to claims of different types. This raises the need for the ability to identify such types. However, the problem is not simply solved by labeling each claim. The reason why context is so important is because the type of a claim depends on the context of its use.

In SBD, a prominent distinction is made between the problem and design domains. In the problem domain, designers explore the reasons for a new system through requirements analysis. In the design domain, practitioners begin to create actual design features to satisfy requirements. Claims can be used in both the problem and design
domains. Problem claims depict artifacts that are deficient within the current method of accomplishing a task. Design claims are created as solutions to problem claims, portraying the future design. A claim could be a problem claim in one system’s context, but a design claim in another.

Norman (1986) presents a cognitive engineering model describing how users cycle through two key obstacles during interaction: the Gulf of Execution and Gulf of Evaluation. During the Gulf of Execution, a user interacts with the system to carry out a determined action. The Gulf of Evaluation outlines how users assess the current state of the system. Similarly, claims can be written to describe what may occur in either of these two gulfs.

The problem and design domains and the two Gulfs create the basis for interactive system design in SBD. A claim can be defined by which two of these four domains it fits into. Its type, however, can change depending on the context. For example, a certain claim may be a problem claim in the Gulf of Execution, but in a different usage context be a design claim in the Gulf of Execution. The key problem is that a claim fits into two of four domain types. Such cases illustrate the importance of context, making it imperative to be able to distinguish between these types of claims. Designers must be able to quickly identify which two domains a claim fits into without having to analyze the details of the whole claim and make a judgment. Our solution is to incorporate the use of abstraction. Section 5.2 provides more on how we succinctly describe claims to identify their types.

3.2. Requirement 2: Selection of Components

When we think of a traditional search, we imagine ourselves providing a query. The query is typically a keyword and/or a value for an attribute. Once entered, search results are displayed. If the user does not find what is needed, he/she is forced to go back and reformulate the query. There is an inherent problem in this process. The user is forced to anticipate the contents of what they are searching within. Without knowing what is available, users go through the trial and error procedure until they find what they need.

One possible solution to this problem is to make the contents of a library visible to the user. Users of a repository must not be forced to guess what is available, but should rather be able to browse through the contents in order to develop an understanding of what the repository can provide them with (Godin, Pichet, & Gecsei, 1989).

However, simply allowing users to browse a library is not enough. As designers browse, they may often find that an artifact does not quite fit their needs. How does one search for an alternate artifact that is close to the general theme of the artifact already found? There is a need to maintain the established design context and also be able to move within the context. To find the most appropriate claim, looking at claims that are within the same design context will give the designers a higher chance of finding what they need. Therefore, searches should build upon previous searches to develop the context the designer needs. Maintaining an established context during the search for a claim should yield better results in terms of finding the claims the designer needs.
Through the process of selection, designers should be able to discover new claims and lead to further alternatives that may be of use. In Section 5.3 we describe how selection allows designers to gain an understanding of a claim and other claims that are within the “vicinity” through networks of claims.

3.3. Requirement 3: Specification to Adapt Components

We can all accept the fact that one can not provide all the components that a designer will need. Even after looking for alternative components, the designer may still not find exactly what is needed. It is very likely the designer will try to get as close as possible to what they need and then stop searching. Because of this, there must be a way to adapt components so that they become fitting for the intended design. This will often lead to the creation of a modified component. The same is true for claims. Designers will want to modify claims they find to suit their own needs during their design process. The challenge is to create the claim correctly and keep it so it too can be reused by others. This is key in creating a cycle of reuse, where new components are created through reuse and then reused by others. Through specification designers can adapt components according to their needs for their own designs. Section 5.4 outlines how designers can specify new claims and include them in the claims library.

3.4. Requirement 4: Integration of Reused Components

It is often difficult to find different artifacts and be able to combine them into a coherent design. While designers are gathering reusable claims, they must begin to think about how all of the claims will interact with each other. It is quite likely that most of the claims they gather will be claims that have previously never been used together. Without understanding how claims will interact with each other, a person is forced to consider a series of ideas without getting understanding how all the pieces come together. There is a need for a representation developers can constantly refer to once major conceptual design efforts have been completed. Additionally, it should represent the current state of the design and provide insight into future design work. The process of collecting and combining the claims is comparable to the process of integration. We propose the idea of a claims map in Section 5.5 to address the issues of integrating claims.

4. CLAIM RELATIONSHIPS

We established the link between problems in enabling claims reuse and Krueger’s reuse aspects in Section 3. Building on the message of providing, “natural, succinct, high-level abstractions that describe artifacts in terms of ‘what’ they do rather than ‘how’ they do it (Krueger, 1992),” we propose the use of relationships as an instrument that will serve this purpose. Relationships between reusable components can provide the succinct descriptions advocated by Krueger.

Previously we identified claim relationships (Wahid et al., 2004a). To implement solutions to the outlined problems, we propose the use these relationships. They show how pieces of HCI design knowledge can be linked together to describe the nature of their connection. Thus, designers can view claims in terms of problems, solutions,
alternatives, sequences, and combinations instead of their internal details. Each relationship is used to succinctly describe the nature of ‘what’ a claim does for an overall design. One can explain the larger implication of a claim for a design, leaving the specifics of what is described within the claim itself for later.

Claims are well-suited for HCI because they describe the psychological effects of design features. Claim relationships must respond to the need for richer descriptions of their use within design processes (Rosson & Carroll, 2002)(Norman, 1986) and their structure. Below, we explain our definitions from Wahid et al. (2004a), but also provide crucial examples we previously lacked: claims and descriptions from actual development project analyses. Two of the relationships we will define are closely tied to the most fundamental process-related design steps followed in SBD. The remaining relationships describe how a claim as a whole or its upsides and downsides can interact with other claims. These two forms of descriptions make the relationships unique to HCI, portraying a plethora of methods to amalgamate HCI knowledge.

4.1. Postulating/Predicating Claims

The first key relationship type between claims is the postulation/predication relationship apparent in the process of mediated evaluation. Design activities in SBD typically iterate through three processes, from requirements analysis to general activity design to specific design of features. In each process, a designer collects evidence to assert postulating claims to guide the next process, while alleviating or refuting claims from the previous process with predicating claims based on new ideas or evidence. Each process acts as a problem that the next process must solve. Thus, there is a natural relationship between the problem and design solution.

<table>
<thead>
<tr>
<th>Maintaining awareness of open windows through a taskbar</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Allows users to recognize that a window is currently open</td>
</tr>
<tr>
<td>+ Enables users to get information at their own will</td>
</tr>
<tr>
<td>- Does not provide a count of the number of windows that are open</td>
</tr>
<tr>
<td>- Requires that the user know that taskbar items are associated to windows</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relaying information through changing text</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Allows more information to be displayed in limited space</td>
</tr>
<tr>
<td>+ Constantly updates user on the current status of information</td>
</tr>
<tr>
<td>- The amount of text visible at any point in time may be too little for a user to understand the message</td>
</tr>
<tr>
<td>- The rate of the change in text may be too fast, not allowing users to read or see</td>
</tr>
</tbody>
</table>

Figure 2. Two claims that have a postulation and predication relationship between them.

To better illustrate the postulation and predication relationships, consider the two claims in Figure 2. The first claim depicts a problem situation describing the current method of maintaining awareness. We use the second claim to solve the problem in the
new design. Thus, the first claim has a postulation relationship with the second. A predication relationship exists in the opposite direction. This association is critical to exhibiting the problems and design solutions that are identified by designers during the development process.

4.2. Executing/Evaluating Claims

As previously mentioned, Norman (1986) presents an argument for interface design as a cognitive engineering discipline, where designers assist the user with progressing through stages of action. Each stage falls within one of two Gulfs—the Gulf of Execution (where the user executes after deciding upon goals and specific action sequences) and the Gulf of Evaluation (where the user appraises the current state of a system). The SBD methodology describes how information design decisions influence the stages of action required for crossing the Gulf of Evaluation, and how interaction design addresses the Gulf of Execution (Rosson and Carroll, 2002). In information design, interface choices such as the use of color, animation, visualization techniques, and layout are made about specific features. Interaction design is more concerned with selection of controls, widgets, affordances, and input techniques.

Claims can be created to specifically fit a certain Gulf. Certainly, a given artifact may be the subject of both Gulf of Evaluation and Gulf of Execution claims. It is helpful to have a relationship to describe this linkage. Some artifacts may only support the user in one of the Gulfs, but may typically be used with other artifacts that address either the same or opposite Gulf. Therefore, the relationship between two feature claims can be described according to the “destination claim.” A destination claim in the Gulf of Execution can be the executing claim for claims in either Gulf. Likewise, a claim in the Gulf of Evaluation could be the evaluating claim for others in the same or opposite Gulf. If desired, a potential task flow can be explicitly represented using the relationships.

<table>
<thead>
<tr>
<th>Recognition of taskbar items for information delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Spacing between taskbar items allows identification of a single piece of information</td>
</tr>
<tr>
<td>+ Flashing taskbar items allow users to quickly recognize the taskbar</td>
</tr>
<tr>
<td>- Flashing taskbar items can cause unwanted interruptions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clicking on a taskbar item to switch active windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Aids in the switching primary tasks</td>
</tr>
<tr>
<td>+ Can increase comprehension</td>
</tr>
<tr>
<td>- Causes a greater primary task interruption due to reallocation of attention to new primary task</td>
</tr>
<tr>
<td>- Previous active window may completely disappear</td>
</tr>
</tbody>
</table>

Figure 3. A Gulf of Evaluation claim and a Gulf of Execution claim.

In Figure 3 there are two claims that fit into the two Gulfs. The first claim belongs to the Gulf of Evaluation since it describes a situation in which the user monitors the state of the interface to receive the updates. The second claim is a direct interaction method and
consequently is a Gulf of Execution claim. When moving from the first claim to the second, we have an execution relationship. The evaluation relationship exists in the opposite direction.

4.3. Generalizing/Specifying Claims

Claims can have different scopes depending on the granularity of the artifact they describe. A general claim might describe psychological effects that result from the holistic design or several distinct portions (combinations of widgets) used in a variety of contexts. General psychological effects can be elaborated by claims that have a narrower scope. These claims apply to very specific parts of an interface (a particular button), usage instances, or user characteristics. They are most useful in guiding component reuse, since they describe an interface at its finest detail and raise in-depth issues related to the interface. However, the underlying theme of a specific claim will often have more frequent applicability to new design problems.

Sutcliffe and Carroll (1999) propose a factoring method for evolving between claims of different scope and use the terms “parent claim” and “child claim.” In our framework of claim relationships, the generalization/specification relationship is the linkage between two claims with different scopes (see Figure 4). A generalizing claim is the consequence of taking a specific claim and generalizing it to apply to a courser artifact or usage context granularity. A specifying claim is the opposite, in that it is the result of narrowing the scope of a general concept. The process of generalizing allows one to create claims applicable to many situations. This course of action permits one to take ideas from a specific problem and reuse them in a new context to solve design issues—perhaps sowing the seeds for innovation and technology transfer. A key concern in generalizing and specifying new claims is with extending or narrowing the scope in an invalid manner, thus, losing the support of empirical or theoretical evidence grounding the original claim. For example, a generalizing claim can only be reliably used in a narrower context, as it inherits upsides and downsides characteristic to specific conditions.

<table>
<thead>
<tr>
<th>Relaying information through changing text</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Allows more information to be displayed in limited space</td>
</tr>
<tr>
<td>+ Constantly updates user on the current status of information</td>
</tr>
<tr>
<td>- The amount of text visible at any point in time may be too little for a user to understand the message</td>
</tr>
<tr>
<td>- The rate of the change in text may be too fast, not allowing users to read or see</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycling banner for relaying text in desktop secondary displays</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Allows more information to be displayed through cycling in a limited amount of space on the desktop</td>
</tr>
<tr>
<td>- Transitions between information may be too distracting due to secondary displays being in view most of the time</td>
</tr>
</tbody>
</table>

Figure 4. An example of a general and specific claim.
4.4. Translating Claims

Existing claims may not be directly applicable to new design problems. They may describe problems that are similar, but not completely accurate. Often though, existing claims provide the basis for the generation of new claims due to recognized similarities between the current problem domain and the one in which the original claim exists. The relationship from the original claim to the new claim is called translation. Translating claims are typically alternatives to each other with the same scope. The relationship may be used to indicate where cross-domain reuse occurs in the development of a system.

Use of fading between information transitions
+ Decreases interruptions caused by dynamic information changing
- Transitions between information may be too distracting if transitions are too fast

Using scrolling to display new information
+ Allows the same amount of space to be used to show more information
- The speed of the scrolling may be too fast, hindering reading
- Transitions between information may be too distracting

Figure 5. An example of two claims with a translation relationship.

The crux of translating is the establishment of a correlation between the existing claim and the claim to be created. To accomplish this, the designer is required to consider the existing claim at a deeper level of abstraction, or a generalized version of the claim. While no explicit generalized claim is created, as suggested by Sutcliffe (2000), the general form of the original claim exists in the mind of the designer. Then, the specific aspects of the original claim are altered to fit its new context of use, thus creating a new translating claim. Ideally, many of the original tradeoffs will still apply in this new context, however, situating the claim requires reevaluation of upsides and downsides with respect to this context.

4.5. Fusing/Diffusing Claims

The fusion relationship between claims is the outcome of the combination of two or more claims into a new fusing claim. A developer recognizes certain aspects of various claims can be applied together in a new and innovative way. The result is a hybrid claim that is pieced together with artifacts and design rationale from each of the supplemental claims. In addition, further design rationale may be required due to novel application of the original artifacts.

Similarly, a designer could break a claim into smaller claims, taking only a fraction of what exists in the original claim to produce a diffusing claim. This time, the designer focuses on part of a larger claim and elaborates on artifacts and tradeoffs that pertain to the new, smaller claim. This practice may result in the creation of multiple smaller claims, depending on how the original claim is divided (i.e. there were equal acting parts
of the original claim). This relationship between the original super-claim and the resulting fractional claim is called diffusion. Figure 6 shows an example of two claims that are fused together to create a new claim.

Mouse-over changes color of clickable text
+ Allows user to understand they may click on the text
- May imply that less information is shown, leaving most of the information to be shown after a click

Red highlighting for urgent or important information
+ Allows users to quickly recognize urgent information, drawing the user’s focus
- Requires a systems ability to determine what information is considered urgent

Mouse-over changes color of clickable text to red showing inaccessible information
+ Allows users to understand whether they should click on the text
- Forces user to mouse-over the text in order to gain the information regarding the status

Figure 6. The first two claims are fused together to create the third claim. The third claim can be diffused into the first two claims.

Relating claims in this manner can illustrate progress throughout design iterations as well as where claim reuse has occurred. Another result of this process may be the fusion of two claims that seem to demonstrate strong positive results in combination or the diffusion of a claim that exhibits distinctively different results for different aspects of its makeup. Additionally, two existing claims from completely different domains may be fused into a new and innovative claim. This process was noted, but not named by Carroll and Kellogg (1989).

4.6. Mitigating Claims

The strength of a claim relies on the explicitness of its upsides and downsides. Upsides represent the potential strengths of an interface, while downsides dictate adverse consequences resulting from the interface design. Explicitly identifying weaknesses of a design often expedites improvement of usability—a process that should be repeated as new flaws are uncovered.

A mitigation relationship is the result of a process in which a new claim is created to manage limitations of another claim. The method of creating mitigating claims can be repeated as many times as needed until designers are satisfied. After designers make improvements to an interface in a design iteration, usability testing must validate the improvements by testing the performance of the mitigating claims. Thus, mitigating claims become a trace of the design improvements that are made over time. Figure 7 shows an example of a claim whose second downside is mitigated by the second claim.
Cycling banner for relaying text in desktop secondary displays
  + Allows more information to be displayed through cycling in a limited amount of space on the desktop
  - Transitions between information may be too distracting due to secondary displays being in view most of the time

Use of fading between information transitions
  + Decreases interruptions caused by dynamic information changing
  - Transitions between information may be too distracting if transitions are too fast

Figure 7. The downside of the first claim in mitigated by the second claim.

5. A THEORY OF REUSE ASPECTS AND CLAIM RELATIONSHIPS

How do we use the claim relationships to enact Krueger’s cause? Claim relationships can be used in different ways to manifest solutions to the requirements described in Section 3. Depending on the way they are used they can exhibit characteristics of each reuse aspect. The use of certain relationships can help abstract claims to identify their type. The relationships can be used to form a network of claims to aid selection. When needed, a designer can create a new claim based on a single relationship. Finally, claims can be integrated together using the relationships. In this section we articulate a theoretical approach to facilitating reuse.

5.1. Claims Library

In recent years, research within HCI recognized that to facilitate reuse, claims must be generalized, classified, stored in a design knowledge repository, and retrieved when appropriate for use within a new design context (Sutcliffe, 2000) (Sutcliffe and Carroll, 1999). Within this literature, approaches for repositories of claims (referred to as claims catalogs or claims libraries) have been introduced.

Although our goal is to prescribe a methodology that can be applied to a reuse repository, we feel compelled to briefly introduce our own repository in an effort to bring clarity to work we did and examples we provide. Our claims library is a collaborative environment in which designers can contribute claims from previous or ongoing designs or reuse claims for ongoing and future designs. All the claims in the library are focused on one class of systems: notification systems. Such systems are designed to deliver valued information without introducing unwanted interruptions to the primary task (McCrickrard et al., 2003). The purpose of the claims library is to provide meaningful design knowledge and encourage designers to search for and consider claims about notification systems similar to an application they are currently designing. The library contains approximately 230 claims. When possible, claim relationships are used to connect claims to each other, forming a network of claims. We intend to only acknowledge the claims library as a test bed for further investigation into our reuse
methodology. Therefore, we will not concentrate on the details of the claims library itself except for those that are relevant.

5.2. Abstraction

Abstraction is the first aspect of reuse to consider. Although it is a very simple notion, one needs to incorporate its use to avoid forcing designers from continuously having to read the details of a complete claim (Krueger 1992). Designers must be able to identify the correct type of claims crucial to the high level design of their system. Through this identification process, they can begin to establish a sense of context within their search process. To tackle this problem of reuse, we assigned relationships to claims in the library. While viewing the details of claims, we allowed users to see the title of related claims and the relationship type. This strategy created a network of claims designers can traverse through by following relationships, a key element in demonstrating the use of abstraction.

In Section 3.1 we described four main domains that form the basis of a design created through SBD: problem, design, execution, and evaluation. During the design process, a designer using claims to create a system should first identify all the problem claims in the Gulf of Evaluation and the Gulf of Execution through the process of requirements analysis. They should then proceed to find all the design claims for both the Gulfs based on the identified problems. This overall procedure requires that they find the appropriate type of claim for each domain when using the library. We adapt these four domains and use them as the basis for abstraction by allowing users of the library to identify an appropriate claim type. In abstraction, it is important to see only the most important data and hide the rest of the details (Krueger, 1992). In this case, when looking at a claim, the only data a user sees for the related claim is the claim title and the relationship type.

The postulation/predication and the execution/evaluation relationships are important relationships when it comes to supporting abstraction in the claims library because they show distinctions between the four domains described in our first requirement. For this reason, they are referred to as the high level relationships. These relationship types allow the user to identify which domains a claim would fit in. For example, if a user looks at a claim’s postulating relationships, they will be able to understand that within the current context the claim they are looking at is a problem claim and that the related postulating claims are design claims. This same idea applies to the execution/evaluation relationships. Designers will be able to understand what Gulf the claim belongs to. Through this simple process, a designer can quickly begin to recognize and locate the correct type of claim that is needed as they simultaneously progress through their design processes (Wahid et al, 2004b). A user study investigating this aspect (described in further detail in Section Error! Reference source not found.) provides some indication of designers being able to identify claim types within an established current context.

5.3. Selection

Unfortunately, as with most knowledge management systems, acquisition is a bottleneck (Wagner, 1990). Component selection is a very important characteristic of
reuse (Krueger, 1992) (Dusink & van Katwijk, 1995), however, the current state of the retrieval mechanisms in such reuse repositories is inadequate. Searches for components are often limited to keywords and classification schemes which only serve as parameters to enhance retrieval. Browsing capabilities to navigate from one component to another may be nonexistent or inadequate in such systems. Furthermore, most design knowledge repositories do not support an outlined search strategy, a series of steps one can follow depending on their needs to ensure that they will find all of the components they need.

Generally, users either search or browse digital libraries (Blanford, Stelmaszewski, & Bryan-Kinns, 2001). A prominent application of the relationships is to use them to aid selection by creating a browsing mechanism (Wahid et al, 2004b). While this attempts to solve our second problem of showing what is available in the library, it also strives to establish a search strategy for finding appropriate claims and alleviate the acquisition bottleneck.

Krueger (1992) describes the selection aspect of reuse in terms of three key concerns of selection: classification, retrieval, and exposition. When selecting, a user must first be able to classify the components. This is similar to understanding the type of the component. Once classified, the user must retrieve the component and then expose the details of it. We proceed to describe how claim relationships can be used to browse our library. A study evaluating the use of selection is presented in Section Error! Reference source not found.

CERVi Search Strategy

Before claim relationships were fully integrated into the claims library, to explore the selection aspect of reuse, the Claims Exploration of Relationships Visualization (CERVi) tool was created to improve knowledge acquisition through the use of claim relationships. Similar to how the relationships are currently implemented in the claims library, the tool allows designers to find claims by navigating through a network of claims connected by claim relationships. In the tool’s case, users are using a visualization of the claims library. Users can find appropriate claims by analyzing related claims, providing recommendations based on the context of the visualized claim.

The tool is designed to enhance the browsing experience and to move away from the traditional idea of searching for claims using a query. During the design process, designers following the SBD approach typically locate relevant problem and design claims for both of Norman’s Gulfs to create a well-planned design. CERVi shows promise in allowing designers to identify problem and design claims and placing them correctly within the Gulf of Execution or the Gulf of Evaluation. Since the process of reuse starts with abstraction, the postulation/predication and execution/evaluation relationships, or high level relationships, naturally form the basis of a search strategy. They enable designers to navigate between problem and design domains and between the two Gulfs. The remaining relationships, called low level relationships, instead allow one to focus within a certain area of the design to find more relevant claims. The overall strategy for browsing supported by CERVi is to first use the high level relationships to
find claims that may fit into the overall design and then use the low level relationships to find the most appropriate claim.

5.4. Specification

Certainly our claims library cannot provide all the claims that are needed for any notification system design. Its contents are finite. It is possible users of the claims library may find a claim that is close to their needs, but not exactly appropriate for their design, forcing the designers to create their own claims when they see fit. Since this is inevitable, the question of how one can then facilitate future reuse of the new claim arises. Important in this investigation is to understand how claims are created and what relationship comes into play.

The act of adapting a reusable component to fit one’s needs falls under the reuse aspect of specification (Krueger 1992). Thus, our third application for claim relationships is to use them as a specification method. Instead of creating a completely new claim, we suggest that the claim should be created by basing it on a claim that is reasonably close to the needs of the design. As the new claim is being created, a relationship should be chosen to guide the creation.

For example, if a designer finds a claim about the general use of color in notifications, they may choose to use it in their design. Of course such a general claim may be somewhat appropriate, but not the best fit. In this case, the designer may wish to use a more specific claim about color—a claim about using the color red to alert users for instance. When creating this new claim, the designer would in effect be creating a specifying relationship from the original claim to the new claim. If this relationship is explicitly noted when creating the new claim, the designer can easily modify the style of the claim, add the new claim to the claims library, and assign the specifying relationship (along with the corresponding generalizing relationship in the opposite direction). This same process is also possible with the high level relationships. When a designer finds a problem claim that is appropriate, but does not find a suitable corresponding design claim, he or she will have to create their own claim. Creating a design claim and keeping the postulation/predication relationship in mind allows the designer to tailor the writing of the claim and assign that relationship when adding the new claim into the library. This is the partial reuse of a claim for the purposes of creating new claims with the ultimate goal of incorporating the most appropriate claim possible into a design.

5.5. Integration

So far we have described how one can identify the correct type of claim through abstraction, browse through a network of claims to select, and specify new claims to adapt. We must now discuss how the claims can be connected together by the designer—an issue addressed by the last reuse aspect: integration. Integration is the process in which a collection of selected and specialized components are combined to form a system (Krueger, 1992). As part of the creative design process, designers using the claims library will most likely have claims that have never been used together before. The aggregate of all the claims they collect will somehow represent the design they want to
create. All the claims must be integrated into a design such that every claim’s role within the context of the whole design is understood. We must also be able to understand what the final result of this process means. A designer should gain value from not just integrating single claims, but from the combined state of all the claims.

How these claims should come together and interact with each other is the final challenge the designer must face. Without an overall understanding of the final conceptual design, designers will find it harder to agree on the most important aspects of the system. We propose the final application of claim relationships be a method for integrating claims toward providing an enduring record of design decisions reflecting how and why claims were integrated in the design process.

Two claims can be integrated with each other when a relationship is assigned to connect them. This behavior can be continued until a group of claims is linked to each other. The structure of claims and the nature of claim relationships allow us to uniquely identify benefits of integration. First, the subsequent group of claims permits a designer to outline high level concepts and strengths and weaknesses of the design being worked on. The collection of claims formed as a result of selection and specification may contain general claims describing overall goals and specific claims which describe system features. When integrated, the goals are explicitly associated through a relationship to the features that enact them. Advantages and disadvantages are expressed in the tradeoffs of the claims and the relationships used to connect them. For example, mitigation shows a solution to an identified problem. The second benefit is the ability for such integration to demonstrate opportunities for testing and redesign. When viewing the integrated claims one can gain an understanding of what is lacking. They indicate when solutions to certain weaknesses do not exist through the absence of certain integrations. This holistic view may even cause designers to determine groups of claims to be insufficient—a possible catalyst to an analytic study by experts. A third gain in integrating claims is the ability for them to represent a trace of motivating factors for iterative improvements over time. Over multiple iterations more claims may be selected or specified and then integrated with preexisting claims.

We refer to a group of integrated claims as a claims map (see Figure 8). Claims maps are directed graphs of claims showing how all the claims collected for a design will work together, representing a decomposition of the whole design. The claims are connected to each other using the claim relationships. Although an explicit hierarchy is not necessary, it is possible for a designer to choose to do so. Claims maps may be a gateway to portraying the three benefits of integration and act as a tool for designers to negotiate design concerns.

5.6. Practicality

Until this point we have covered the components of a methodology to further reuse using relationships. There is a large concern to implementing such an approach. Can such a model actually work and become sustainable? There are a number of issues to consider if we are to fully deploy this methodology. We must critically analyze how it will work and its likelihood of success.
One concern is populating repositories such as the claims library to the point where one can find reusable components that are of use. Without a core set of claims, a claims library will not provide value to designers. This will require a set of people familiar with a certain domain to be willing to spend time creating claims that can address the common concerns of the domain. An immediate benefit of such an effort is the recording and documenting of knowledge representing the current state of the domain. This can give others the opportunity to familiarize themselves with the area and potentially contribute to its advancement. On the other hand, we understand this will require a certain degree of personal sacrifice on the part of the individuals who spend time creating claims. One way to approach this is to try to focus to the contents of a repository to a manageable class of systems so that contributors do not spend time creating content for everything they imagine.

One potential advantage of using relationships is that they can establish a cycle of reuse. By cycle of reuse we are referring to abstraction and selection allowing designers to find claims and specification encouraging the creation of claims. The key is to capture the new claim with a relationship and connect it to the rest of the claims so that others can find it. But how can we get people to reuse, contribute, and connect claims? Concentrating on how a person acts within the context of a design activity is the solution to how we can accomplish this—a notion that diverts our attention to the use of claims maps.

One possible direction to take is to focus designers on the claims maps instead of the claims library. Deploying this approach would require tools that can provide a claims map environment overlying a claims library. This would provide two main advantages. First, when claims are created and linked in the claims map, the same claims can be stored in the library and connected together using the same relationship. This would allow others to find the claims in the future. A community of users that create claims maps can add to the content of the library. This strategy can prove useful because contributions to the library are made for a purpose and not because of good will. Claims would be linked together to create a design and not just to maintain a claims library. The other advantage arises when a person is expanding their claims map. If they are using a claim that was collected from the claims library and placed in the claims map, this approach also allows other claims to be suggested. For example, if a claim needed to be mitigated the person could check to see if the same claim in the library has a mitigating claim connected to it. This would allow designers to share the claims submitted by a community and build on them by incorporating them into their own work.

This vision may be the kind that takes a step toward transforming a reuse methodology like ours to one that functions transparently. Focusing on creating claims maps and providing access to the library makes users shift between abstraction, selection, specification, and integration without knowing it. With appropriate tool support we can eventually shift from a few people populating a repository to a community of users that contribute, share, and use claims. We must note, however, that our intention is not to focus on the sustainability of this vision by specifying the details of the tools that are needed since it is beyond the scope of the paper.
6. EXPLORING THE UTILITY OF CLAIM RELATIONSHIPS

To this point the paper has described a conceptual framework describing how one may facilitate reuse—leveraging Krueger’s four tenets of abstraction, selection, specification, and integration—through a claims creation and relationship identification process. We have argued how our expanded definition of claims provides a foundation for capturing design knowledge and how claims relationships facilitate understanding and reuse in the design process for each of Krueger’s tenets. This section exemplifies and explores how claims relationships and our associated methods can support design reuse through three controlled investigations.

The first study investigates how our framework for abstraction and selection leads to a reasonable understanding of claims’ contribution to a new design. This is done through the navigation enabled by the use of relationships. The second study examines how novice student designers can create claims by specification, leveraging at least four of the claims relationships. This is an important issue to investigate due to the fact that a library’s contents are finite and may not always yield what is needed. The third study investigates the integration of claims—both new and reused—into a complete design by encouraging the development of claims maps by small groups of graduate student designers. We focus on the process of creating a claims map, the type of structure that is incorporated, and its ability to allude to further testing and redesign.

6.1. Exploring Abstraction and Selection

According to Krueger, adoption of the first two tenets of reuse—abstraction and selection—ensure that designers can appropriately abstract existing claims to new design problems and select the appropriate claim for each aspect of the design. Our first study probes to what degree designers can demonstrate not only abstraction, but also an appropriate degree of selection in matching claims to design problems using claims relationships. Our goals were to determine the impact of claim relationships on finding reusable knowledge, providing an initial degree of confidence in the use of relationships in the abstraction and selection processes. We tested this by asking designers to create an initial system design by searching for and collecting claims that would support a design problem. A usable abstraction and selection approach should result in designers collecting appropriate claims and placing them in the correct context, thereby improving the base of design. (A broader presentation of this study and its results can be found in Wahid et al., 2004b); in this section we present a more in-depth examination of the results with closer consideration of the abstraction and selection tenets.)

Participants

Six senior HCI graduate students participated in this study. All of them were familiar with claims through courses and their own graduate research work, though none of them previously had been exposed to claim relationships (i.e., none were directly working on our research effort). Our desire for experience in the use of claims limited us to a small number of participants, but we feel this level of experience was necessary to adequately
reflect the ability of designers to understand claims written by others and match them appropriately to stages of design.

Materials

Study participants were tasked with matching design claims to the appropriate stage of design, according to SBD’s six stage adaptation (Rosson & Carroll, 2002) of Norman’s seven stages of action (Norman, 1986). A design table served as a tool to outline the conceptual design in terms of claims. The table consisted of two rows, one corresponding to the problem domain and the other to the design domain, and six columns, each representing one of Norman’s adapted stages of action. The intention was to write claim ID numbers into each cell of the table when chosen for the design. While each of the claims were appropriate for one or two stages of action, they could not deviate too much from their intended stage of action (e.g., a claim that would fit into a Gulf of Execution stage would not apply to any of the Gulf of Evaluation stages).

A slideshow tutorial was developed as an introduction to claim relationships to define each one and explain how they can be used to search. A sheet containing instructions, definitions of each relationship, and a scenario describing the need for a notification system in a train station was provided to each participant. A short survey was prepared for the study, first asking participants to describe their design vision, then asking questions related to Krueger’s (1992) definition of selection and its application to reuse. All questions were geared toward exploring our claims and claims relationships in supporting selection (see Table 1). Answers were recorded using a 5-point Likert scale, ranging from Strongly Disagree to Strongly Agree, with optional written responses. The questions addressed general selection issues as well as classification, retrieval, and exposition concerns within selection.

Procedure

The students were presented with the ten-minute slideshow tutorial and were given the instructions, relationship definitions, design table, and scenario for the study. The students were asked to create a high-level design of a notification system that would satisfy the needs outlined in the scenario by collecting claims from the claims library and recording the claim ID numbers in the design table. They were to search for claims using the claim relationships using a claims database. Because relationships form a non-hierarchical network of claims, it is often hard to find a starting point within the network. For this reason several claim ID numbers from the claims library were included under various stages in the design table to initiate a search for claims, while other portions of the table were left blank. We also gave participants the option to use a traditional search with text queries to find other such starting points if they felt they were not satisfied with the given starting points. Once they finished their designs, they proceeded to complete the survey. They articulated the type of system they envisioned and then answered all the questions relating to selection. During the duration of the study they were allowed to ask us questions regarding definitions or the design table, but they could not ask questions regarding what claims to use.
Results and Discussion

We calculated the number of errors that were made in the design table based on a predetermined solution set. The solution set was not created with a single design in mind, but instead considered all the possible claims that were relevant to a specific location in the design table. Thus, our solution set allowed us to perform a syntactic check on the design table to understand if claims were placed in the correct context irrespective of the actual design itself. For each design table we recorded the number of correctly placed claims, claims in the wrong Gulf, claims in the wrong stage in the correct Gulf, and claims in the wrong domain. We then calculated a design score and percentage for each participant, with a 1.0 weight for incorrect Gulf and domains and a 0.75 weight for correct Gulf but incorrect stage identification (recall that claim relationships cannot be used to distinguish among stages). In general, the small sample sizes preclude us from exploring statistical significance—we report here on statistics, trends, and observations from our investigation.

Four of the six student designs scored over 80%, with an overall average design score of 72.86%, and a standard deviation of 0.23. (The two low scores were 63% and 30%.) A majority of the students’ claims were placed correctly in the table. The participants demonstrated an understanding of which Gulf each claims should be placed in by using claim relationships. Most errors were due to incorrect identification of stage within the correct Gulf, a reflection of the incapability of relationships to distinguish between stages. The two low scores suffered from both Gulf and stage errors—perhaps indicating a lack of understanding.

Gulf and domain errors are indicators of participants’ inability to abstract using the high-level relationships (see Section 5.2). The scarcity of Gulf errors indicates users can indeed successfully abstract using the high-level relationships. Errors regarding the stages do occur much more frequently, no doubt because claim relationships do not directly help to distinguish between stages, as noted previously. However, the ability to identify the correct Gulf should increase opportunities of identifying the correct stage are increased. These initial results serve to demonstrate the role of abstraction and selection in the claims reuse process, allowing for the placement of claims in the correct context.

To perform a semantic check on the designs we qualitatively assessed the design tables and concluded four of the designs had claims sufficiently related to the scenario and stated design visions. A set of domain experts (including authors of this paper) created a predetermined solution set containing all possible claims we judged to be appropriate for the given scenario. Claims were judged to be appropriate if they appeared in our solution set and were related to the declared design vision. In the event a claim did not appear in our solution set, we relied on the design vision to judge whether the participant did choose a claim that would suit their design. We acknowledge there may be an inherent bias in our judging because we did not opt to use an independent set of judges due to the early nature of this study. However, we believe our predetermined solution set served to minimize such bias.
Even though participants had the option of using a regular text search, none used it—possibly indicating the use of claim relationships may result in the retrieval of appropriate claims given that the claims are available.

Table 1 summarizes the results of the survey. Responses to the classification questions, related to the conceptual structure of the relationships. Although we can not point to any statistically significant results with just six participants, various metrics indicated the group had a reasonable understanding of the distinction between high and low level relationships after the introductory tutorial. Responses to the retrieval questions, related to the design of CERVi, showed that participants could easily retrieve a claim based on a displayed relationship. Answers to questions related to exposing claim details indicated examining claim details was a key aspect of the selection process.

<table>
<thead>
<tr>
<th>Question</th>
<th>Concern</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>I was able to locate useful claims using the relationships.</td>
<td>General Selection</td>
<td>86.7% Strongly Agree</td>
</tr>
<tr>
<td>I was able to understand why two claims had the relationship they had.</td>
<td>General Selection</td>
<td>73.3% Agree</td>
</tr>
<tr>
<td>I used the given claim IDs as an index for locating more claims.</td>
<td>General Selection</td>
<td>90.0% Strongly Agree</td>
</tr>
<tr>
<td>The relationships gave me an understanding of what the related claim does.</td>
<td>Exposition</td>
<td>83.3% Strongly Agree</td>
</tr>
<tr>
<td>The claim’s details were key in the selection of the claim.</td>
<td>Exposition</td>
<td>76.7% Agree</td>
</tr>
<tr>
<td>The relationships allowed me to understand where a claim could be placed on the table.</td>
<td>Classification</td>
<td>80.0% Strongly Agree</td>
</tr>
<tr>
<td>I was able to easily retrieve a claim based on a displayed relationship.</td>
<td>Retrieval</td>
<td>80.0% Strongly Agree</td>
</tr>
<tr>
<td>I understood the distinction between higher level and lower level relationships.</td>
<td>Classification</td>
<td>73.3% Agree</td>
</tr>
</tbody>
</table>

Table 1. The questions asked related to specific characteristics of selection and the results of both rounds.

The results of our evaluation suggest that our approach of using relationships has promise in helping designers improve designs, as the application of claim relationships for selection provided a valid and useful means for finding claims for most participants. Through the use of relationships, users gained an understanding of the library contents without having to otherwise anticipate through the creation of search queries. Furthermore, the identification of the correct context allows designers to find appropriate claims, a need specified by our first requirement.

6.2. Exploring Claim Selection

Our previous study showed claim relationships facilitate navigation and identification of potentially applicable claims. However, because a claims library can not be exhaustive, we must also consider how one can build upon the ideas that are found in the
library. Thus, a remaining question is whether relationships can support and characterize claim creation. Having a standard claim creation process can define what should be done to create a claim using a particular relationship, ensuring a higher quality of reusable claims. Our intention was to investigate the possibility of an emerging standard claim creation process based on relationships. We decided to run a class activity in which we asked students to create claims using given relationships.

We identified four trends in creating claims for each of the low level relationships. We anticipated that the use of the generalization/specification relationship to create a claim should preserve the underlying concept of the original claim, but have new upsides and downsides related to the claim feature. The fusion relationship should result in the new claim having all the tradeoffs of the base claims in a synthesized form, but a new feature. The diffusion relationship should only retain some of the applicable tradeoffs. Using the mitigation relationship should result in a new claim with a feature describing the solution to a particular downside in the base claim. Translating a claim should yield a claim with a similar feature with some tradeoffs being preserved.

Participants

Seventy-eight undergraduate HCI students took part in the study. They were taking their first introductory course in HCI. Prior to the study they had been exposed to claims in the class and had completed several assignments and activities using claims.

Materials

The students were given a set of 10 claims (some are in the explanations of the relationship types) representing techniques for the design of a news ticker that relays news headlines from the corner of a desktop monitor. To aid them, they were provided with written definitions and examples of each relationship type. They were also given papers that indicated what relationship they should use to create claims and forms to record their claims. A short slideshow was prepared to deliver the instructions for the activity.

Procedure

Once the students were given the materials, we proceeded to deliver the instruction during a short presentation. The students were asked to choose any two claims from the set and create two more claims based off of those claims. Half the students were asked to use the generalization/specification and translation relationships to create the new claims. The other half was asked to use the fusion/diffusion and mitigation relationships. This distinction was indicated in the papers they were given.

We did not place any restrictions on time and the types of claims that should be created, permitting us to observe what claim creation would be like if our whole concept were to be fully deployed. They were free to ask us any questions regarding instructions. Students were free to either create a claim that fit well with the system described by the claims or create a completely new claim that did not necessarily relate to the system. Our intention was to mimic how contributions to a claims library could potentially be made.
Claims written for reuse may be generic enough not to relate to a specific design or specific enough to only function in specific situations. The lack of any such restriction allows us to analyze many different types of claims. The use of the postulation/predication and execution/evaluation relationships for this study was avoided because we felt it would force many students to think solely within the bounds of the given system.

Results and Discussion

For the generalization relationship students identified words or concepts that could be described in more general terms. For example, students would generalize red by referring to it as color. When specifying a claim, students tended to add adjectives to make certain conditions more specific. For both the relationships, the generalization and specification of the claim title were imperative. New upsides and downsides were created for both these relationships, but some were taken from the original claim if they still applied. Although we anticipated the creation of new upsides and downsides, we did not expect as many adoptions of original tradeoffs.

During the fusion process, a new claim encompassing all the claims being fused was created. The title usually contained concepts from all the claims being fused. Most of the students took the definition of fusion literally, combining the upsides and downsides from all the claims into one claim. When diffusing, students identified a single concept and usually created a completely new claim. The fusion process was somewhat consistent with the behavior we predicted.

Mitigation and translation both resulted in the creation of a completely new claim with new upsides and downsides. While creating a mitigating claim, students made sure the title described a solution to the chosen downside. Translating claims were created by first identifying a word or concept in the original claim and then identifying an alternative concept for the new claim. The importance placed in the formulation of the title when mitigating was expected by us. This demonstrates the value placed in the title of claims themselves as another form of a succinct description. We did not foresee the creation of completely new translating claims. We believed many of the new translating claims would share many similarities due to the fact the relationship connects alternative claims to each other. Instead, the students took the chance to create alternative claims that were of a completely different nature with fairly different tradeoffs.

As we were not able to conclude the existence of any distinct claim creation process for any of the relationships with statistical significance, we can not firmly accept any of our hypotheses. However, the behavior of students tended to meet many of our expectations for the use of claims relationships.

The nature of this study is to analyze the semantics of new claims. Since it was not a test of HCI knowledge per se, we feel comfortable in speculating what expert designers (not just students) might do. Experts may in fact be able to create new claims using relationships, using their valuable insight for the formalization of claim creation in the cycle of reuse. More importantly, the study serves to illustrate ways that relationships
can be interpreted differently, freeing designers to creatively investigate alternatives. The reuse of claims, and parts of claims, is an important catalyst for creation, as designers use the structure of claims—not only the definitive parts like upsides and relationships but also semantic parts like nouns and verbs—in the creation process.

6.3. Claims Map Creation

Our previous studies revealed that claims have promise as a starting point in understanding abstracting, selecting, and specifying features of existing designs toward planning future designs. We speculate that claims maps, introduced in Section 5.5, provide the interface overview necessary to integrate the concepts in a designer’s mind. To explore the concept of claims maps we sought to observe how claims maps were created for a system. We coordinated a class activity in which we asked students to create claims maps. Our goals were to determine how claims would be integrated together, whether the claims maps would outline system goals and features, and depict areas for future testing.

Participants

This study was conducted with 18 graduate students who were taking a graduate level usability engineering course. Their experience in HCI varied depending on whether they were engaged in HCI research for their own work and the number of years spent in graduate school. They were introduced to claims earlier in the course and completed a previous assignment on claims.

Materials

Students were referred to a paper on SideShow, a notification system that docks to the side of the screen to provide updates on information such as weather conditions, number of software bugs, and online buddies (Cadiz et al., 2002). Students had printed copies of the paper, descriptions of each claims relationship type, and a screenshot of the SideShow system was projected for the duration of the activity. Participants used their own sheets of paper to develop their claims maps and blackboards to draw their final claims map.

Procedure

Prior to the beginning of the in-class study, the students were asked to individually create their own claim outside of class based on an aspect of SideShow. Our reasoning for asking the students to create claims individually first stems from our desire to mimic an actual design session in which one would collect disparate claims from a claims library and integrate them together. Once the students arrived in class, they were divided into four groups, with 4-5 in each group, and asked to create a claims map for SideShow. They were told to examine design decisions made in the paper regarding SideShow toward understanding how the original designers captured claim relationships and how an explicit representation of them can help future redesign. We did not introduce any manipulations in the design of this study because we wanted to expose all the students to
claims maps and because our goal was to analyze the different types of claims maps that could emerge.

Students proceeded to present the claims they created to each other and then started discussing what should go into their claims map. We asked that each claims map have at least 8 claims and use at least 3 relationship types. To ensure the activity would be completed within an hour we limited the relationships to only the lower level relationships. We permitted them to ask us any questions if they needed clarification regarding instructions or the use of relationships. At the end of the activity, groups drew their claims map on a board to show to everyone and turned in the sheets with their claims map and full claims to us.

Results and Discussion

Our first intention was to determine how the claims were integrated together. We expected claims would need to be modified and adapted during the integration process. Our method of asking participants to create claims individually proved successful because each group had a diverse set of claims which described different aspects of the same system. Because students were taking different claims and trying to integrate them, we expected they would have to create new claims or alter some of their preexisting claims. Indeed, the students did so to bring other claims together. On average, each group created 4 more claims to have a claims map suitable enough to represent the system. We provide an example of one of the claims maps developed by a group which had four members (see Figure 8). Not all the claims initially created by students were used. Some claims were too similar to use both. Other claims could not fit into the structure the groups tried to create in their claims maps. While extracting claims leads to less sense of ownership than creating claims and designs from scratch, our approach was intended to create and resolve conflicts among the designers.

We believed claims maps would be used as a vehicle to understand and outline system goals and features. This phenomenon was indeed exhibited in the types of claims that were used and created. General claims, such as the claim about management and organization of information in Figure 8, depicted high-level goals addressed by the system. They outline motivations for building the system. Specific claims describing the features were identified and integrated with the high-level claims, proposing solutions for the goals. The advantages and disadvantages of the system are encapsulated in the various tradeoffs of the claims. A mitigation relationship in the example claims map points out a specific weakness of the design that was addressed by another claim to strengthen the system. Through quick recognition of such design characteristics, one can notice the potential utility of claims maps.

Crucial to the notion of claims maps is also their ability to depict opportunities for testing and redesign. These chances are normally instantiated when new claims are created, downsides are left unaccounted for, and relationships need validation. Thus, we dissected the claims maps to understand whether such insights would be found in the students’ claims maps. Many of the concepts encapsulated by the claims were derived from the literature about the SideShow system, however, many others were created
during the integration process, initiating a need for testing. The new claims and their tradeoffs represent aspects of the design that should be tested to validate the effects of the feature. For instance, the video feed claim in the example was created as a result of a specification to demonstrate the system’s ability to deliver dynamic information. This was not specifically addressed by the SideShow publication (Cadiz et al., 2002), but instead was an observation made by the students. Certain claims may need small analytic studies while other groups of claims may require longer field studies. Further opportunities for redesign arise when downsides are unaccounted for (through the use of mitigation). Significant downsides can represent potential risks to the design of the system, requiring immediate attention. Determining the magnitude of certain downsides is a catalyst to conducting further testing. The validity of certain relationships can also be a point of contention if they are not evaluated. Is the claim in Figure 8 about using icons truly an alternative to providing short summaries? What is it about these two claims that make them alternatives to each other? Does the collapsible groupings claim really mitigate the problem of hidden information? These are pressing questions situated within the nature of the relationship used for integration that can only be answered through evaluations.

Figure 8. A claims map about the SideShow system created by one of the participating groups. Many claims are from literature (Cadiz et al., 2002) while others were derived using the relationships. Questions about the validity of certain claims and relationships (indicated by the dashed line) and claims not mentioned in the paper, but created by students (shown by the dotted lines), should be answered through testing.
The accumulation of opportunities for testing further motivates the use of claims maps as an evaluation tool. The map can be given to expert evaluators to interpret the state of the design and determine where further redesign should occur. These observations may situate claims maps as a tool to gauge the state of the design and trigger evaluations and redesign efforts when needed.

Claims maps may establish a possible format for creating a full fledged design record with proper tool support, allowing designers to truly gain value. We envision a tool that tracks the development of a claims map over time, showing when claims were integrated. Design rationale explaining the additions or even the removal of claims can provide further insight into the development. Tracking of tested and untested claims will allow designers to monitor the state of evaluations. With the three benefits showing some promise, we speculate that such tools will eventually show promise in helping designers abstract, select, specify, and integrate design elements of others in their own designs.

7. CONCLUSIONS

We presented a framework for how HCI professionals can develop, access and make use of a body of reusable design knowledge. We argue that this framework can lead to a way for HCI professionals and developers to build on and continuously improve HCI design knowledge. We focused on the methodology for enabling reuse rather than the contents of our specific claims library. Our goal of shedding light on reuse through the work we have presented has been based on two important concepts: Krueger’s (1992) aspects of reuse and claim relationships. Using relationships for abstraction and selection allows designers to identify and retrieve the correct types of claims from a repository. Claims creation through specification supports designers who need to tailor claims for their own work. Finally, claims maps enable designers to integrate claims and consider the low level details of how their design should really function. This work provides three main contributions to those in HCI who wish to explore the methods for how to reuse and design:

• Presenting a case for reuse within HCI and providing an example of how a software engineering perspective is relevant to the problems we face in HCI regarding the reuse of design knowledge.

• Introducing an approach for design knowledge reuse in HCI that instantiates Krueger’s vision of reuse through claims and claim relationships.

• Examining key components of the methodology and presenting evidence that the approach supports a framework that could enable design reuse in HCI.

Our approach advocates the use of design rationale throughout the design process. Claims are a form of design rationale that provide insight into the reasoning for why an artifact could be used. Claim relationships and claims maps are extensions that aim to provide more rationale for the design. As Horner and Atwood (2006) mention, design rationale systems can be very useful for reflecting upon decisions, communicating design issues, and analysis of previous decisions, but state they can suffer from limitations in
capturing, using, and organizing rationale. There is a need for tools that support our framework and address these limitations.

Claims capture information that can impact the use of artifacts, but are only as good as the information they record. A useful claim will depict accurate, novel, and interesting feature tradeoffs. Claims that list obvious tradeoffs or do not list critical and important tradeoffs will be less useful for subsequent system developers. However, when first developed, claims may not capture this important information. Like software systems, claim quality can be improved over time through iterative enhancements. Within this context, social systems may be beneficial in improving and maintaining claim quality. Leveraging a community’s ability to determine information that is useful may prove effective in filtering out or improving claims.

As a design representation, claims maps capture one holistic perspective of the system design. There is a need to establish and manage how they can be used in conjunction with other design information such as scenarios and graphical representations. One possible option is to record design information in a linked claims map and storyboard structure, allowing for the capture of rationale, look, and system flow. Further work is needed to explore and compare the costs of maintaining these representations versus the utility they provide over the course of a design project, and look at how tool support can mitigate those costs while increasing utility.

Another reuse issue that needs to be explored is how well claims can apply to different design contexts outside of those in which they were intended for. When using a claim in a different context, there is an inherent tradeoff between how detailed the claim is and how generic it is. The more generic a claim is, the more it is likely to apply to many different situations, but because of the lack of details someone trying to use the claim might find it to be lacking. One way to address this problem may be to organize claims into layers of detail and provide a tool that allows designers to move between levels of granularity to allow for better navigation and application of general and specific claims to different contexts.
REFERENCES


Henninger, S. (1997) An evolutionary approach to constructing effective software reuse repositories. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 6, 2, 111-140


Ralph E. Johnson, Frameworks = (components + patterns), *Communications of the ACM*, 40, 10, 39-42


