Entering the Heart of Design: Relationships for Tracing Claim Evolution

Shahtab Wahid, C. F. Allgood, C. M. Chewar, D. Scott McCrickard
Center for Human-Computer Interaction and Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0106 USA
{swahid, callgood, cchewar, mccricks}@vt.edu

Abstract
Designers need guidance in tracing knowledge to support the iterative development of interactive software interfaces. Claims show promise in capturing design knowledge with concise descriptions of an artifact's psychological effects on users, but adoptions and modifications made during design processes result in new claims. The manner in which new claims are created based on previous claims establishes unique knowledge relationships not well captured by existing research. This paper proposes six claim relationship types presented with general concepts and examples, allowing a more robust claims analysis process to emerge. The definition of relationships acknowledges claim evolution methods inherent in design, facilitating knowledge reuse and providing structure to advance the science of design.

1. Introduction

The advancement of HCI as a science hinges upon the transfer of knowledge over time within the field. Just as important as the transfer method (design approaches, reuse paradigms, etc.) is the form and structure of the knowledge itself. What better to encapsulate this knowledge than *claims*, design rationale that unapologetically captures both the good and the bad of an artifact? The claim structure proposed by Carroll describes the psychological effects of a designed interface artifact in a usage scenario [1][2][3][10]. Claims address a variety of situational and interface aspects that affect the compatibility of the design and user's models, such as user satisfaction and feeling of reward, color and object layout, and strength of affordances. Inherently objective, claims provide designers with an unadulterated view into what makes an artifact live and breathe, grounded in theories and observations of user experiences.

To illustrate the concept of a claim, we consider a generic timeline artifact that could be used to view all activities and deadlines related to a project. Timelines have been used extensively in information management, resulting in numerous broad statements about their usage summarized as an example claim in Figure 1. This construct concisely illustrates the tradeoffs of using a timeline with the upsides and downsides of the claim. Through design research and innovation, we try to preserve the upsides and mitigate the downsides.

Timelines that dominate the organization, monitoring, and filtering of data . . .	
	+ add historical context and aid temporal logic by organizing work, correspondence, and transactions in the order that they occur
	+ provide a natural guide to experience as a universal skeleton-key
	− BUT can subsume metaphors suggested by other interface artifacts and hierarchical categorizations
	− BUT may add to confusion by giving an improper timestamp to data with ambiguous temporal characteristics

Figure 1. Claim about a timeline artifact, from [4].

Therefore, it is important to examine claims as they change and evolve, are created and reused. Just as there are a multitude of human relationships as new generations are born and they themselves reproduce, we propose that there are many claim relationships that exist during the development and evolution of design artifacts. Recording and understanding these relationships provides deeper insight into the overall design process.

Why is this important? Recognizing claim relationship types during the design of an artifact impacts both current and future designs. It enhances the current process by providing a more detailed view of the design history so better decisions can be made during future iterations. Claim relationship types supply valuable knowledge of a claim's origin and development for its potential reuse in another context. In summary, explicit relationships aid in the transfer of claim-embodied knowledge in both the short-term and the long-term.

A need exists for a rich set of classifications for claim relationships. We propose six new claim relationship types in this paper. We also illustrate how our relationship types may come to exist during the development of a system, building from the timeline example above. These

relationship types are by no means exhaustive or all-encompassing. Rather, they are the beginning of a new perspective on the depths of claims and a science of HCI.

2. Related Work

Claims analysis supports the practice of mediated evaluation [9] in human-computer interaction and provides process for evolving a record of design rationale, an argument introduced more than a decade ago [3]. As a form of mediated evaluation, claims analysis blends the benefits of intrinsic evaluation (where a design is described in terms of the performance characteristics it supports) and payoff evaluation (where success in meeting design goals is determined near the end of a project)—it allows explicit and deliberate goal formation, testing, and revision early and often throughout the course of design. As an evolving record of design rationale, the set of claims forming a claims analysis is a series of hypotheses and observations about an artifact in use. While potential benefits have been recognized for making and reusing claims [10], formal and complete guidance for describing relationships among claims is not available.

Claims are one component in Carroll's task-artifact framework [1] and scenario-based design process [8] that helps designers recognize tradeoffs implicit in the design as users form a goal, act toward its achievement, and evaluate progress. Articulating these tradeoffs as useful generalizations for future design work provides a mechanism for generative problem-solving and design, integrating theory development with design evaluation [3]. Based on the task-artifact framework and the notion of claims reuse, Carroll and Sutcliffe have developed a gradient of progressively powerful object-oriented design analysis techniques whose potential can only be realized with a more clearly defined claim structure [1][10][11].

Certainly, other approaches to design knowledge reuse are prevalent in the software engineering community, especially patterns and object modeling. In our thinking, claims are compatible with both the HCI processes embedded in scenario-based design and patterns records—with *claims as the heart of a pattern (from an HCI perspective) and the focus of usability engineering work*, expressing the key psychological tradeoffs of the reusable artifact modeled by the pattern. Claims-to-pattern relationships are likely to be a many-to-one.

3. Claim Relationship Types

We propose six new claim relationship types that respond to the need for richer descriptions of claim structures and iterative processes within claims analysis. This section defines each relationship type in turn using general concepts, while Section 4 illustrates integrated relationships in a working example.

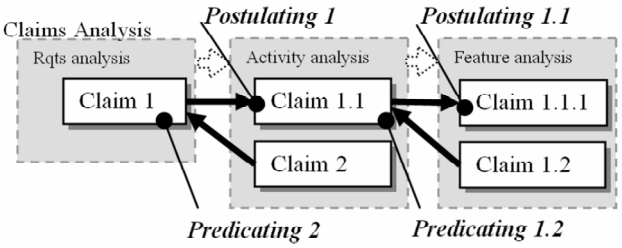

Figure 2. Postulating/Predicating Claims

3.1. Predicating/Postulating Claims

The first key relationship type between claims is the *predication/postulation relationship* apparent in the process of mediated evaluation. In a claims analysis, a designer assigns credit or blame attributions to artifacts, which are continuously refined in subsequent design activities. Design activities typically iterate through three processes, from requirements analysis to general activity design to specific design of features—a pattern paralleled by the themes addressed in each claims analysis. *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.*

As illustrated in Figure 2, a designer would make Claim 1 to express aspects of the problem domain based on requirements analysis. This leads to the creation of Claim 1.1 as a potentially valuable new user activity through postulation. As specific interface features are conceptualized (Claim 1.1.1) to support the desired user activity (Claim 1.1), Claim 1.1.1 can be referred to as a postulating claim of Claim 1.1. All of the claim upside and downside tradeoffs could be elaborated with scenarios, illustrated with storyboards or other prototypes, and tested with users. Through these design development processes, designers gain inspiration about new ideas—here, an alternate feature (described by Claim 1.2) is found to offer better support for the activity described by the predicating claim, Claim 1.1. Likewise, proposed or validated activity concepts (Claim 2) would be predicated by a claim about the problem domain (Claim 1).

Relating claims in this manner preserves their role within an evolving design rationale context. Recognizing claims in a role as open propositions provides an impetus for continued design development and testing. Alternatively, antecedents or propositions backed by solid evidence suggest a potentially reusable design artifact.

3.2. Executing/Evaluating Claims

Norman presents an argument for interface design as a cognitive engineering discipline, where designers assist the user with progressing through stages of action [7]. He

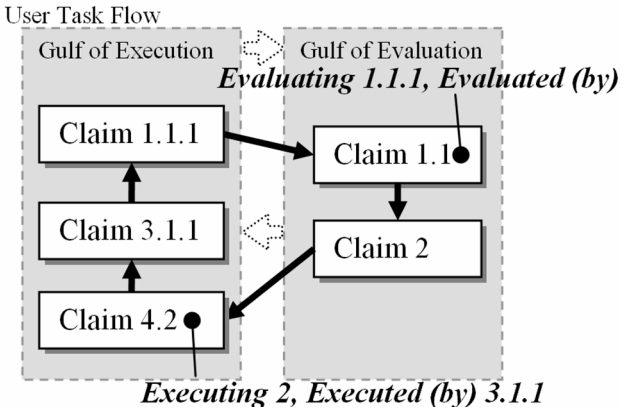

Figure 3. Evaluating/Executing Claims

describes two key hurdles within the stages—crossing the *Gulf of Execution* (after which goals and specific action sequences are decided upon) and the *Gulf of Evaluation* (where the user appraises the current state of a system). Rosson and Carroll's scenario-based design 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 [8]. In information design, interface choices such as 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.

Certainly, a given artifact may be the subject of both information and interaction claims, and it would be helpful to have a relationship to describe this linkage. Other 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 other claims in the same or opposite Gulfs.*

The user task flow determines where the execution and evaluation relationships exist between claims. For instance, if a user's task flow involved Claim 1.1.1, then Claim 1.1, then Claim 2, and so on (see Figure 3), the claim relationships could be described as follows: Claim 4.2 is the executing claim for Claim 2 and is further executed by Claim 3.1.1; Claim 2 extends Claim 1.1 by elaborating evaluation features; and Claim 1.1 is the evaluating claim for Claim 1.1.1. To preserve the context of the task flow, the chain of claims should be related as precisely as possible (for instance, Claim 2 should not be described as an evaluating claim for Claim 3.1.1, without including the intermediate links).

Having a simple vocabulary to describe the relationship of claims across the stages of action and Gulfs operationalizes Carroll and Kellogg's notion of "task coverage" [2]. As a heuristic for sufficient detail in a claims analysis, *task coverage* is achieved when at least one claim describes each major artifact state within the task flow across the Gulfs of Execution and Evaluation. In later work, Carroll specifically cautions against replacing a single artifact or claim within a series of task coverage claims, lest the context of task flow be broken [3]. As we move toward developing libraries of claims for reuse, keeping execution and evaluation relationships explicit will preserve task context and assist designers with establishing task coverage in claims analyses.

3.3. Generalizing/Specifying Claims

Claims can have different scopes depending on the granularity of the artifact components which 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 "general idea" of a specific claim will often have more frequent applicability to new design problems.

In our framework of claim relationships, the *generalization/specification relationship* is the linkage between two claims with different scopes. *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 specializing 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 (see Claim 2 in Figure 4). This course of action permits one to take ideas from a specific problem and reuse them in a new context to solve design issues—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.

Sutcliffe and Carroll propose a factoring method [10] for evolving between the two types of claims mentioned, although they use the terms "parent claim" and "child claim." This process involves an analysis of the claim and the situation in which it is used, and allows production of new claims from existing claims. The method is used to examine how a claim's generalized form spans different

contexts. In the context of this method, since one analyzes a specific claim in order to generate a general claim, the parent is the specific claim and the derived (general) claim is the child claim. Unfortunately, the terms are misguiding. With Sutcliffe and Carroll's terminology, a specific claim that leads to the creation of a general claim would be described as "a child spawning a parent." The terms do not distinguish between directions the scope of a claim can change, motivating our argument for the use of generalization and specification relationship types.

3.4. Translating Claims

Existing claims may not be directly applicable to new design problems. *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.* Ultimately, claims linked via the translation relationship indicate where cross-domain reuse has occurred in the development of a system (e.g., translation from Claim 1 to Claim 2 in Figure 4).

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 [11], 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 re-evaluation of upsides and downsides with respect to this context.

3.5. 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 that certain aspects of various claims can be applied together in a new and innovative way, such as Claim 3 in Figure 5. The result is a sort of 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 (e.g., Claims 2.1 and 2.2 in Figure 5). 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.*

Relating claims in this manner can illustrate progress throughout design iterations as well as where claim reuse has occurred. During the design process, testing and evaluation provide the basis for the validation or alleviation of claims. 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 problem domains may be fused into a new and innovative claim. This process was noted, but not named by Carroll and Kellogg [2]. An intermediate step, similar to the generalization process described above, requires the designer to consider "what" the claim does, as opposed to "how" this is accomplished. This distinction depends on the level of abstraction at which the claim is considered. In this instance, fusing claims is similar to integration as described by Krueger [6]: the designer "must clearly understand . . . those properties of the artifact that interact with other artifacts." This is accomplished by considering an abstract version of the claim "in which the internal details of the artifact are suppressed."

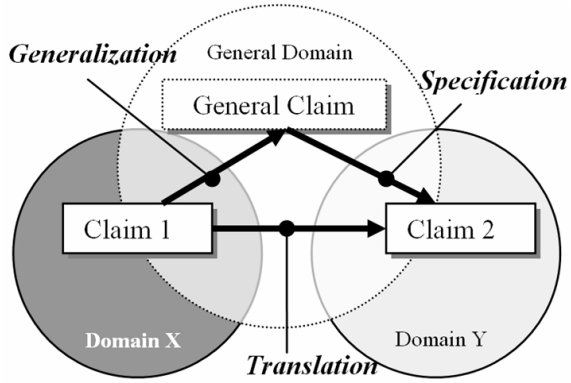

Figure 4. Generalizing/Specifying, Translating Claims

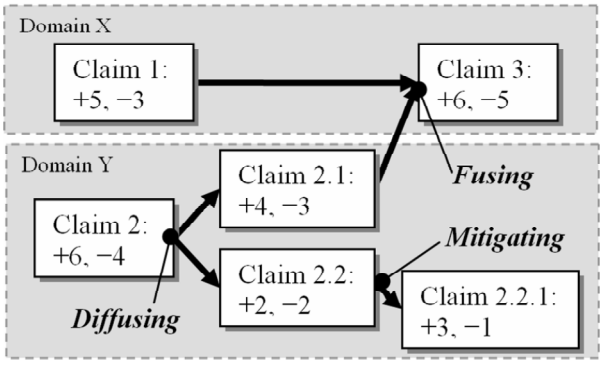

Figure 5. Fusing/Diffusing, Mitigating Claims

3.6. Mitigating Claims

The strength of a claim relies on the explicitness and poignancy of its upsides and downsides. Upsides can represent the potency 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.

Scenarios are descriptions of a sequence of mental and physical actions a user of an interface may go through. Carroll suggests that one can use scenarios in order to construct new alternative scenarios [1]. The process of analyzing the psychological design rationale within a scenario allows designers to identify alternative scenarios which may be appropriate for other possible usage scenarios. Alternate scenarios are created in a way so that they can handle or correct disadvantages and at the same time maintain or improve strengths of other scenarios.

This same process is valid for claims. *A mitigation relationship is the result of a process in which a new claim is created in order to manage limitations of another claim.* As previously mentioned, claims make their downsides explicit, clearly identifying areas for which designers must also find solutions. The purpose of a mitigating claim is to resolve the downside in order to improve the overall design (Claim 2.2.1 in Figure 5 removes a downside and gains an upside on Claim 2.2). 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 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.

The repetition of mitigating claim creation and testing for verification produces a chain of mitigating claims. Each claim mitigates a downside in the previous claim. In such a chain, solutions to problems can easily be found, helping general reuse. Typically, the beginning of the chain may contain solutions to slightly more general problems. As more specific problems are identified, mitigating claims find solutions that are more specific. A claim that is further down the chain may turn out to mitigate, not only the claim used to create it, but claims that are even higher up the chain.

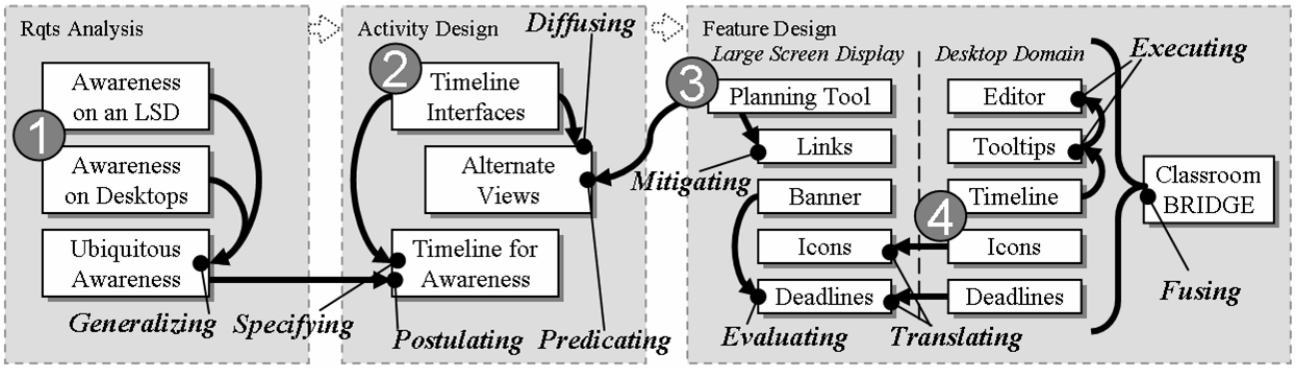

Figure 6. Claim relationships for the ClassroomBRIDGE project; boxes represent claims referenced in Section 4.

4. Example—Claim Relationships in Design

To illustrate how claim evolution takes place in usability engineering efforts, we turn to our development of ClassroomBRIDGE, a collaborative project management tool for middle school science classes [5]. ClassroomBRIDGE built on several previous efforts, both internal to our group and drawn from other researchers, making it rich with examples of claim evolution.

One of the first requirements was developing interfaces that could be used by students at their desk and teachers throughout the classroom. The central technological addition to our suite of classroom tools was a large screen display, positioned at the front of the room. Even though the interfaces were used in different ways—students constantly study the desktop systems, while teachers quickly get guidance from a large screen—we recognized that both requirements could be expressed as a *generalizing claim* for ubiquitous awareness (see "1" in Figure 6). We *postulated* that using a timeline metaphor for activity awareness supports both user requirements.

In initially brainstorming appropriate activity design approaches, we were intrigued by the Timeline Claim (see Figure 1). We realized that our idea of a timeline metaphor suggested a *specifying claim* regarding the utility of timeline displays expressed in the original Timeline Claim. However, we did not wish to employ the full power of the Timeline Claim as put forth by the authors; instead we created a *diffusing claim* in which we maintain many of the upsides of timelines yet still provide alternate views to the data (see "2" in Figure 6).

Extensive use of a prior, similar system developed by our group revealed limitations in our overall approach—

predicating claim downsides of the alternate view implementation would apply, creating usability concerns. Specifically, the student interface contained a planning tool with the downside: created pages were rarely viewed and never updated after creation. We *mitigated* this downside in our new interface with links in the timeline to the planning tool pages. The timeline links provided a constant reminder of recently added pages, encouraging review and update by the students (see "3" in Figure 6).

In designing specific features (see "4" in Figure 6), we realized that the large screen display would provide teachers with a constant progress view of all student teams on a timeline similar those on student computers. However, our multi-platform system also necessitated that we support many system elements, like the timeline, on both desktop systems and the large screen display. As many elements of the desktop systems were already created and tested, we had to translate much of the information to the large screen display, often reusing elements like the work artifact icons and deadline markers. This was done by making *translating claims* from desktop systems to the large screen for each artifact.

As we discussed previously, the timeline view is not the only view available to students. One challenge in building ClassroomBRIDGE was in connecting the timeline to a concept map, notification banner, chat tool, editor, and other views. Our solutions resulted in numerous *evaluating, executing, and fusing claims*. For example, we color coded related elements in different views to bridge the gap between perception and interpretation in users, two stages in the Gulf of Evaluation and the basis for one of our *evaluating claims*—deadlines were shaded with yellow in both the notification banner and the timeline view. As a second example, to assist users with forming new action plans and initiating execution within the timeline view (Gulf of Execution stages), we implemented tooltips showing authors and dates of work items that would launch appropriate tools when clicked. The tooltip *executing claim* would help a user initiate an update action for a document they recognized to be out of date.

5. Conclusions and Future Work

We have proposed a framework in which claim relations can be named and described as claims evolve over time. A lot of previous work has been done on claims, but little has focused on claim relationships. Our work fills a need for identifying and defining types of claims and links that may exist among this reusable design knowledge.

The primary purpose of such definitions is to make explicit an individual claim's role within the larger claims analysis and derivation across multiple design studies. Since an interface is the aggregate expression of many claims working together, each claim establishes relationships with other claims. The six relationships we define allow designers to more richly describe claims in the widest context possible by describing relationships to other claims. By enabling a record of claim evolution, our framework permits one to understand the process used to derive a new claim or reuse a claim in another domain.

Our future work consists of developing a tool based on this framework to organize a claims analysis. This visualization will show all the claims being used in an interface development process along with relationships to claims in a design knowledge repository.

We envision our framework as not only being able to describe *pro forma* design rationale, but to provoke reflection and creative thought processes that would not otherwise be explored by designers. Many of the implicit processes used to generate new claims may be innate for experienced designers, but this formalism will be valuable for design education. With the many benefits of our claim-type definitions, we lay the foundation for a science of design within human-computer interaction.

References

[1] Carroll, J. M. Making use: a design representation, *Communications of the ACM 37*(12). December 1994.

[2] Carroll, J. M. and Kellogg, W. A. Artifact as theory-nexus: Hermeneutics meets theory-based design. In *Proceedings of the Conference on Human Factors in Computing Systems* (CHI'89), ACM. New York, 1989. 7-14.

[3] Carroll, J. M., Singley, M. K., and Rosson, M. B. Integrating Theory Development with Design Evaluation, *Behavior and Information Technology 11*, 1992.

[4] Freeman. Eric and Gelernter, David. Lifestreams: A Storage Model for Personal Data. SIGMOD Record (ACM SIG on Management of Data *25*(1), 1996.

[5] Ganoe, C., Somervell, J., Neale, D., Isenhour, P., Carroll, J. M., Rosson, M. B., and McCrickard, D. S. Classroom BRIDGE: Using Collaborative Public and Desktop Timelines to Support Activity Awareness. In Proceedings of the ACM Conference on User Interface Software and Technology (UIST '03), Vancouver BC Canada, Nov 2003.

[6] Krueger, Charles W. Software Reuse, *ACM Computing Surveys (CSUR) 24*(2), June 1992.

[7] Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper, Eds. *User Centered System Design*, 31-62, Hillsdale, NJ: Erlbaum.

[8] Rosson, M. B. and Carroll, J. M. (2002). *Usability Engineering: Scenario-Based Development of Human Computer Interaction.* Morgan Kaufmann Publishers.

[9] Scriven, M. (1967). The methodology of evaluation. In R. Tyler, R. Gagne, & M. Scriven (Eds.), *Perspectives of curriculum evaluation*. Rand McNally, 39-83.

[10] Sutcliffe, Alistair G. and Carroll, John M. Designing claims for reuse in interactive systems design, *International Journal of Human-Computer Studies 50*. 213-231, 1999.

[11] Sutcliffe, Alistair. On the effective use and reuse of HCI knowledge, In *ACM Transactions on Computer-Human Interaction*, *7*(2), 197-221, June 2000.