Education and Design: Using Human-Computer Interaction Case Studies to Learn

Laurian Hobby, D. Scott McCrickard
Department of Computer Science and Center for HCI, Virginia Tech
Blacksburg, VA. 24061-0106
{lhobby, mccricks}@cs.vt.edu

ABSTRACT
Computers are essentially an ever-present tool that can be used in almost any discipline to make work faster and easier. Creating these programs, however, such that they fulfill the needs of the customer is a challenging process given the uniqueness of the discipline and circumstance. Thus, the use of a programming design methodology can enable the computer program designer to create a better system that meets the needs of the customer. Teaching this process, or in essence how to design, is the focus of this work. In this paper we present how using case studies in Human-Computer Interaction, and more specifically displaying the evolution of a case study, increases a designer's ability to learn and then apply this knowledge. We investigate how to use this design evolution within case studies and the effects it had on application, while also exploring how case studies can be used in educating computer scientists.

Author Keywords
Human-computer interaction, case studies, scenario-based design

INTRODUCTION
How do you teach design? Even with the skills and tools given to programmers, such as languages and compilers, how does a computer system designer know how to make the right design? How to teach the art of creating an interface? These are questions that are being discussed in the Human-Computer Interaction (HCI) community. This paper discusses these issues and explores the use of case studies of successful computer designs as a tool to leverage learning.

A case study can be described as a collection of artifacts or data about a particular example or phenomenon. In the instance of a computer program design, a case study is a collection of the artifacts, skills, and tools used to create the end program. As an analogy, in building a house, the agreements made between the architect and the builders would be one datum and the set of architectural drawings for the design would be another.

Case studies have had a long history of teaching in many various disciplines because of the diverse nature of what a case study can be. Perhaps it is the nature of a case study in that tells a story that is easier to understand than a rote methodology. Or, perhaps its popularity is due to the fact that people can more easily understand and relate to the story-telling and personality of the work. Perhaps they remain popular because case studies are able to show the details between the steps. No matter which of the above is correct, case studies have the ability to make learning, and more importantly *understanding*, easier.

This application of case studies as a teaching aid brings up many other questions. For example, what pieces of information are important to store/archive for others to learn from later? Secondly, what is the most effective way to present the case studies to novice designers to maximize the extent of their learning? Thirdly, how can a case study be developed and when should it be developed? In this work we study what information should be used within a case study. This affects all three of the posed questions and is foundational in making this work successful.

BACKGROUND AND RELATED WORK
Case studies are a form of analogical problem solving; a method in which a user is able to apply the lessons learned in one discipline to a new situation. This method of learning is effective due to the psychological methods that motivate creativity [9]. Specifically, the cognitive mapping of one situation in the mind to another is what makes the use of case studies not only possible, but a normal function for people to use in a new situation.

When using a case study a student will follow characteristic steps in order to solve the presented problem. These steps, according to Aamodt and Plaza, are retrieve, reuse, revise, and retain [2]. First, the student has the need to solve a problem and thus attempts to retrieve relevant material from some library of cases. The student will then select the case based on some set of defined criteria and determine a unique solution using that material. Once the solution is finalized it is archived within the original library [10]. This cycle creates a process of knowledge reuse, augmentation, and propagation, which assists the community to be successful while using case studies.

In the HCI designer community there is a tendency, however, to try and "recreate the wheel". Rather than looking through past designs and design decisions, as successfully practiced in business [1] or biology [3, 13], precedence has shown a tendency to be ignored. This could be due to a multitude of factors, two of them being the age of the discipline and the diversity of computer programs to design; however, neither should be allowed to limit the creation of a set of case studies. To be successful in this endeavor, some standards should be applied to what needs to not only be in a case study but how the case study should unfold for the user. Just as a book is made up of sentences, paragraphs, and chapters to tell a story, a case study should provide a structured account of a past experience; and, just as a book has a table of contents, bibliography, and other structural content, a case study should possess items that facilitate easy use. One way to achieve this result is to employ a design methodology which defines a set of standard artifacts and a timeline.

One design methodology that has been successful in educating young HCI students is Scenario-Based Design (SBD) [5, 14]. In SBD designers are encouraged to develop computer systems, with the user in mind, by working with scenarios. A scenario is a descriptive story that tells how an end-user would interact with an interface. In a way, SBD forces the designer to think through various design trade-offs and decisions that have to be made. For example, a notification message can be salient but it can also be overly interruptive. These tradeoffs are called 'claims'. By working closely with the end-user of the interface, the designer will be able to visualize the proposed system early in the design process because scenarios and claims are written in the user's language (i.e., not in engineering jargon).

In terms of a case study, the artifacts created while using SBD could be combined in such that they would aid the illustration of design. An introductory attempt has been lead by Carroll and Rosson to create a case study library [6, 15]. This library[1] is a collection of artifacts (i.e. scenarios, claims, and other artifacts) that follow the SBD design process. Through using these artifacts, teachers are able to design class activities that best promote HCI education. It should be noted that this library currently contains only a half a dozen cases since this is early in its development process. How to design a case study and what information should be used in a case study are harder questions to answer at this early research stage. Currently the artifacts that are presented in the Case Study Library are the easiest ones to collect: the final products. While these may support an overview of the design process they may not highlight the intricacies of the steps. For this fact we are focusing on determining artifacts highlight the design process.

DESIGNING CASES

While visualizing a case study and building a library of cases are important tasks, determining what should be in a case study is more important as this will establish its usefulness. A case study, in terms of SBD, needs to be able to drive class learning activities. Creating the cases such that they maximize the learning, and provide the best reference tool, requires that the artifacts within the case should be carefully explored.

The field of perceptual learning includes the concept that by presenting *contrasting cases* the student will be better able to detect what may be subtle differences between artifacts [8]. This has ramifications for what case artifacts should be present. Currently, one finalized artifact (i.e. the final version of a scenario) is present, but the process and the artifacts used to evolve to this state are missing. Common experience shows that the first draft of any work product is never the best. By presenting the evolving versions of these SBD artifacts, and showing progression down the design process, students will be better able to understand the methodology used, and thus at a higher level the method of HCI design. As an artifact evolves it will better reflect its design requirements, e.g., as a scenario evolves it will better incorporate the needed elements in order to convey the story of use.

In this study we explored how a scenario evolved in order to be able to describe the impact it would have on learning. Wahid et. al. has explored the idea of how claims would evolve and speculated not only how a claim would evolve but also the impact it could have on deign [17]; however, in the literature claims are not as well defined as scenarios which at this point in time makes them harder to evaluate.

In writing a scenario while using SBD there are eight elements that should be included: Setting, Actor(s), Task Goals, Plans, Evaluations, Actions, Events, and Plot (See Table 1). Scenarios should also have certain characteristics that are harder to define: Succinctness, Concreteness, Flexibility, Coherence, and the ability to Promote Work Orientation (See Table 2). As a scenario evolves the writer

Element	**Definition**
Setting	Situational details that motivate or explain goals, actions, and reactions and actors(s)
Actors	Human(s) interacting with the computer interface or other setting elements
Task Goals	Affects on the situation that motivate actions carried out by actors
Plans	Mental activity directed at converting a goal into behavior
Evaluations	Mental activity directed at interpreting the features of the situation
Actions	Observable behavior
Events	External actions or reactions produced by the computer or other features of the setting
Plot	Arrangement of incidence to convey the story

Table 1. Elements of a scenario. (Adapted from [5, 14].)

[1] The Case Study Library can be found at http://ucs.ist.psu.edu/

Characteristic	Definition
Succinctness	The ability to convey the right amount of information about the elements
Concreteness	The ability to be firm in the details about the elements
Flexibility	The ability to leave future adaptation about the elements
Coherence	Logical flow and aesthetic consistency
Promote Work Orientation	Enables all stakeholders to understand the design at any point in the design process

Table 2. Characteristics of a scenario. (Adapted from [4, 10].)

would not only incorporate more elements of a scenario but the writer would also be able to balance concreteness and flexibility in such a way that the scenario would promote work orientation. Teaching these finer elements and characteristics is hard not only for a student to understand, but it is hard for an instructor to evaluate, thus providing contrasting cases could enable a better understanding for the educational process.

EXPERIMENT

Experimental Design

In this study there were two hypotheses: 1) That evaluators will be able to attain a high level of agreement on what a 'good' scenario is; 2) That participants would create better scenarios in the after working with scenarios that showed the evolution of design. To address these hypothesis, three groups of users (n=41; two groups of 14, one group of 13) were employed to create a three factor experiment – two experimental groups and one control group. The participants were all enrolled in a junior year HCI class being taught in the spring semester of 2006. Each participant was given a set of four scenarios and a list of the elements of a scenario with descriptions as provided in Table 1. The participants were then instructed to isolate the elements of a scenario within each of the four scenarios. Next, the participants had to write a follow up scenario based on the previously viewed scenarios. These scenarios were viewed as the final material to be evaluated for this study.

The three factors were control, expert, and novice. In the control group the participants read four different scenarios about the same piece of software. In the expert and novice groups the participants read a progression of the same scenario with the fourth being the final and perfect version. The expert group read the progression of scenarios from the point of view of an expert SBD writer; where as, the novice group read the progressions from the standpoint of a novice SBD writer. Expert scenarios evolve to incorporate and weave more material into the story; novice scenarios, on the other hand, are usually overly verbose and try to incorporate too much or too little. For example, a novice will occasionally incorporate a funny anecdote that is unrelated to the plot: "George got really angry and had to crack open a beer...". Through feedback a student learns that this type of information is unneeded, and it was thus incorporated into the novice experimental factor.

The scenarios produced by the participants were then graded by three evaluators. Each of the fifty-one scenarios was evaluated based on a) the elements of a scenario and b) the characteristics of a scenario; with a double-anchor ten-point scale in which "one" was the best grade and "ten" was the worst. Three evaluators were used, two female and one male. The male and one of the female evaluator had taken one prior SBD design class. The other female evaluator had programming experience but had not taken any HCI classes. The results were then evaluated for inter-rater reliability [12]. Inter-rater reliability is the statistical test to measure how much the evaluators agree on a criterion. For example, if the there judges scores were 6, 7, and 6, this would have a high level of agreement: the scores were similar. For this study the inter-rater reliability was done for each characteristic and element and agreement is measured in terms of percentage. The average for each element was then taken across the three evaluators for the participants final score. These scores were then used to determine the effectiveness of the contrasting cases.

For the inter-rater reliability, the null hypothesis was that the evaluators would have a successful rate of agreement. If a successful level of agreement is found and the null hypothesis is disproved, then this would prove the there is agreement on what makes a 'good' scenario. Given a successful level of agreement for the first stage, the null hypothesis for the participant scenarios was that all three factor groups would perform at similar levels across all elements and characteristics of a scenario. If the null hypothesis is disproved for this stage then it would prove that contrasting cases could be used to help teach design.

Results

Inter-rater Reliability

The interclass correlation between all three groups was very low. Taking the average measures in a two way random effects model – where both people-effects and measure-effects were random – the interclass correlation was 0.31 (df = 50). In contrast, when the rater who had not had any previous HCI or SBD experience was removed, the interclass correlation between the two remaining evaluators was 0.67 (df=50). There is a correlation of agreement between the two judges that could be relied upon. This disproved the stage one null hypothesis and allowed for a continued evaluation of the participants' scenarios.

Participant Scenarios

For the participant scenarios the null hypothesis was disproved: there was a statistical difference between the groups. Using a two-tailed t-test there was a statistically

	Control		Expert		Novice	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
Setting	3	0.91	3	0.91	3	0.92
Actors	3	1.51	2	1.04	2	0.78
Goals	5	1.09	4	0.75	4	0.93
Plans	5	1.03	4	0.89	4	1.13
Evaluation	6	1.52	5	1.06	4	1.31
Actions	5	1.65	3	1.04	4	1.32
Events	6	1.34	5	1.15	5	1.40
Plot	4	1.05	3	1.11	3	0.64
Succinct	3	0.98	3	0.87	3	1.00
Concrete	4	0.99	3	0.69	3	1.12
Flexible	3	0.70	3	1.07	2	0.36
Coherence	3	0.97	3	0.86	3	0.67
Work	2	0.53	2	0.98	2	0.47

Table 3. The mean and standard deviation for the three factor groups based on scenario elements and characteristics

significant difference between the control and the experimental groups – experimental groups being novice scenarios and expert scenarios combined. This means that in the following elements and characteristics of a scenario the contrasting cases groups performed better than the control group: Actors (p=0.049), Goals (p=0.035), Plans (p=0.01), Evaluation (p=0.032), Actions (p=0.04), Events (0.081), Plot (p=0.044), and Concreteness (p=0.061). (See Table 3 for statistics on mean and standard deviation.) It can be seen initially that elements of a scenario can be more easily isolated and transferred to future work by participants. However, characteristics of a scenario may be harder concepts for students to mentally digest. In terms of a difference between expert and novice groups there was a moderate statistical difference in the element of Evaluation with expert scenarios having a better score (p=0.08; Expert Mean=5, Expert SD=1.08, Novice Mean=4, Novice SD=1.31). These results prove that contrasting cases can affect certain scenario factors to help teach SBD to undergraduate students.

Discussion

Looking at the results we can see that the use of contrasting cases, in particular showing the evolution of design, can help educate; in essence, it can help teach people how to design. The use of contrasting cases or even HCI in computer science seems atypical to computer programmers. This may be due to the influence of mathematics where there is one correct answer to a problem. In design this is simply not the case; instead, there may be many possible solutions that all influence the final outcome.

An analysis of how a doctor might treat a patient- by recommending the patient drink a glass of juice every day or by prescribing a dose of antibiotics in order to achieve a healthy state- may be analogous to understanding that there are many solutions to a design problem: determining which method to use is where case studies can play a part in design. When trying to determine which case to use, or how to apply that knowledge to the application of ones own work, being able to see the steps between the starting point and the finalized solution can highlight the intricacies of a delicate process. As we can see from the results shown in Table 3, having students see the design evolution, whether from the novice or expert standpoint, students were able to see the subtle differences between the scenarios and then apply that knowledge to writing their own material.

In terms of the scenario evaluation, it is encouraging to see that agreement about what makes a "good" scenario is possible. Although it takes training in SBD to get a correlated agreement, this is not surprising. These evaluators have had feedback from expert scenario writers in their class instructor. This possibly enabled the evaluators to have a better grasp on the elements and characteristics of a scenario due to their own experience with using SBD. In other studies, when trying to get a significant correlation of agreement, most evaluators have to work closely for a period of time before the evaluation can start. This process, in a way, works to guarantee a larger agreement because of a defined set of rules created by the evaluators – ones that were not given by the experiment proctor. In this study no such process was used: the evaluators worked independently and had not participated in group work together. This means that in general, with previous work in SBD, subjectivity of what makes a good scenario can be limited.

The lack of agreement in the characteristics of a scenario – succinctness, flexibility, coherence, and promoting work orientation – is not surprising either. The characteristics of a scenario are harder to grasp then whether or not an element of a scenario is present. It is similar to asking someone to judge figure skating: the rates of agreement in artistic quality are going to vary more than ones based on technical skill due to the intrinsic biases of what makes good art. These are harder concepts to grasp and are good examples of those that we believe need more work.

The lack of statistical significant for the element of Setting can be explained by looking at the activity within the experiment. In this activity the participants had to follow up on past materials. For seven of the eight elements the participants had to construct new material which showed a difference in their abilities. For the element of Setting the participants were able to copy almost all of the material from the previous work across all three experimental factors. This is most likely what caused a lack of difference for this element.

Lastly, the lack of a statistical difference between expert and novice scenarios is surprising. This may be due to a low population of users; however, the t-tests between the two groups showed that there was little or no difference between the two experimental groups. It is possible that the participants were able to learn the same lesson through different means which caused the same end results. This topic will have to be explored in future work.

Overall, these results make a strong argument for using contrasting cases while trying to teach and/or learn from SBD. Much of the time spent in a class involves the students receiving feedback from a teacher about their work and then turning the feedback into a better product. By showing the evolution of a scenario the student is able to see the application of unspoken – and not illustrated – feedback; this student is then able to apply the gained knowledge to their own future work. Not only does this reduce the costly time spent on providing teacher-student feedback, but it also gives the student a reference tool in guiding their own work. For example, 'Dan wants to better incorporate his actor's motivations and goes to a case study where this is illustrated.' This application facilitates a faster learning time and allows the students to focus on harder areas of design development.

Apart from just a reference tool for students to refer back to this new work should be applied to actual classroom activities. In terms of practical application, instructors should be able to focus class activities on using a case library to design scenarios. In a jigsaw -like activity where students are given intermediate elements, students can explore the previously used scenarios while focusing on one particular aspect. For example, if a classroom of forty students was broken into eight teams, each team could focus on one element of a scenario and how its use evolves while designing. Activities that involve case studies need to be explored in the future.

While SBD may be only one design methodology or process, it shows that students are able to take the story from one set of data and produce a solution that is better. This ultimately means that these students will be able to better convey their designs to stakeholders. Needless to say, this is an important process to learn when designing software solutions – where feedback from the stakeholders is sometimes what will make or break the success of a piece of software.

In the larger picture this work makes a convincing argument for a way or tool to teach design: using case studies. A program manager or a designer will be able to use a case study and a case study library to learn how to make the right design; by following the precedence set forth by past designs programmers will be able to raise the floor on their ability to create a successful design.

FUTURE WORK

The next phase of this work is going to be expanding the contrasting cases to other components of SBD. This will naturally lead to the future development of a case study that will show the evolution for all established SBD artifacts.

Using contrasting cases is only one of the many new theories that might be added to a case study. For example, when trying to determine which claims to use, a student may look through a similar case study to see which claims were discarded or how they were modified. Expanding on this idea, a map of the claims and how they all interact to create a final design is another part of a design case study that should be tested.

Discovering how these cases should be visualized and support the activities associated with learning also needs more development. Some preliminary work has been done using the current case study library; however, the activities are based off of the library, rather than the library and its cases being based on the activities[6]. One piece of information found in the previous work performed on visualizing cases is that the activities for learning are stifled by the current visualization[4]. Participants were not able to apply understanding from one phase of SBD to the next. Creating a library/case that is networked and displayed in such a way that it is intrinsically obvious that it will make the library/case usable and educational will be of great benefit.

In the larger picture of cases, finding a way to make the case library self propagating would be a large stride in the success of this work. At Virginia Tech there is a group that is working on a project that is planned to guide designers through the steps of SBD. This project, called LINK-UP [7], works with a library of SBD claims to have the designer create the needed artifacts of SBD process. For example, as designers write scenarios they may choose claims from the library and incorporate that information into their scenarios. Working to take the created artifacts out of LINK-UP and into a case study would automate the process of creating, leveraging, and modifying cases.

In terms of education, using different types of case studies has been effective in the education of HCI students. The work done by McCrickard, Chewar, and Somervell points out that using a traditional case study library- like the one at present- requires a significant background in the material before application can be possible [11, 16]. The use of familiar interfaces such as AIM or Yahoo was the most productive in enabling students to apply the theory within HCI; however, students felt that the case study library was the best reference to refer back to and use. Entwining both of these educational materials in a case library would make the best tool for young designers.

The current work presented in this paper moves towards the future work to be completed in understanding how to use case studies for HCI education. Determining what should be in a case study is the first step, along with determining what activities to support with a case study. The next step of this work is to determine how to visualize the materials and to propagate the library of case studies.

CONCLUSION

Some people have argued that design is an art and it cannot be taught. If this is true, then why are there art schools and programs? Art, the same as design, is a creative process. Knowing how to utilize that process to yield the highest results is what teaching design is all about. In this paper we have discussed how case studies and how a case study library should be involved in teaching HCI. One element of that is determining what parts or artifacts of design should be presented within a case study.

The research presented in this paper demonstrates that by showing students how design evolves, students were able to grasp a higher understanding of a popular design process. This understanding was then transferred to their own work to produce a better design in eight out of the thirteen graded criteria. This is an encouraging step in trying to answer the question of how to teach design and one that should be explored in future work.

ABOUT THE AUTHORS
Laurian Hobby is a first year Ph.D. student at Virginia Tech. D. Scott McCrickard is her advisor and an Assistant Professor at Virginia Tech.

ACKNOWLEDGMENTS
We would like to thank Jason Lee and Shahtab Wahid for their help with crafting this work and the Virginia Tech Statistical Department for helping with the statistical modeling and testing.

REFERENCES
1. *Havard Business School Cases*. 2006, The website of Harvard Business School Publishing. http://www.hbsp.harvard.edu/products/cases/.
2. Agnar Aamodt and Enric Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, in *Artificial Intelligence Communications*. 1994. p. 39-52.
3. Mark Bergland, Karen Klyczek, Mary Lundeberg, Kim Mogen, Douglas Johnson, Marlys Nelson, Torey Kauth, Sam Cocchiarella, Tim Krehl and Tim Hiller. *Case It! Case-based learning in biology*. 2006. http://caseit.uwrf.edu//caseit.html.
4. Brandon Berry, Laurian Hobby, D. Scott McCrickard, Chris North and Manuel A. Pérez-Quiñones. Making a Case for HCI: Exploring Benefits of Visualization for Case Studies, in *ED-MEDIA*. 2006: Orlando, FL.
5. John M. Carroll, *Making Use: Scenario-based Design of Human-Computer Interactions*. 2000. Cambridge: Massachusetts Institute of Technology.
6. John M. Carroll and Mary Beth Rosson, *A case library for teaching usability engineering: Design rationale, development, and classroom experience.* J. Educ. Resour. Comput., 2005. 5(1): p. 1-22.
7. C. M. Chewar, Edwin Bachetti, D. Scott McCrickard and John Booker. Automating a Design Reuse Facility with Critical Parameters: Lessons Learned in Developing the LINK-UP System, in *Computer-Aided Design of User Interfaces IV.*, Jacob, Robert and Limbourg, Quentin and Vanderdonckt, Jean, Editors. 2005, Kluwer Academic Publishers. p. 235-246.
8. Wendell Garner, *The processing of information and structure*. 1974. Potomac, Md.: Earlbaum Associates.
9. Mary L. Gick and Keith J. Holyoak, *Analogical Problem Solving.* Cognitive Psychology, 1980. 12(3): p. 306-355.
10. Ramon López de Mántaras and Enric Plaza, *Case-Based Reasoning: An Overview.* AI Communications, 1997. 10(1): p. 21-29.
11. D. Scott McCrickard, C. M. Chewar and Jacob Somervell. Design, Science, and Engineering Topics -- Teaching HCI with a Unified Method, in *Proceedings of the 2004 ACM Technical Symposium on Computer Science Education (SIGCSE '04)*. March 2004: Norfolk, VA. p. 31-35.
12. Kenneth O. McGraw and S. P. Wong, *Forming Inferences About Some Intraclass Correlation Coefficients.* Psychological Methods, 1996. 1(1): p. 30-46.
13. Centers for Disease Control and Prevention. *Epidemiologic Case Studies*. 2005, Department of Health and Human Services. http://www.cdc.gov/doc.do?id=0900f3ec80093d70.
14. Mary Beth Rosson and John M. Carroll, *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. 2002. San Francisco: Morgan Kaufmann Publishers.
15. Mary Beth Rosson, John M. Carroll and Con M. Rodi. *Case studies for teaching usability engineering*, in *Proceedings of the 35th SIGCSE technical symposium on Computer science education*. 2004, ACM Press: Norfolk, Virginia, USA.
16. Jacob Sommervell, C. M. Chewar and D. Scott McCrickard. Making a Case for HCI: Comparing Materials for Case-Based Teaching, in *2004 Conference on Frontiers in Education (FIE '04),*. 2004: Savannah, GA. p. T3H-4 - T3H-9.
17. Shahtab Wahid, C. F. Allgood, C. M. Chewar and D. Scott McCrickard. Entering the Heart of Design: Relationships for Tracing Claim Evolution, in *Proceedings of the Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE '04)*. 2004: Banff Alberta Canada. p. 167-172.