EXPERIMENT ON A SYSTEM-LEVEL DESIGN TOOL

Joshua A. Ruder and **Durward K. Sobek, II**
Mechanical & Industrial Engineering Dept.
Montana State University
Bozeman, Montana 59717-3800
Tel: 406 994 2203
Fax: 406 994 6292
dsobek@ie.montana.edu

September, 2005

Abstract

System-level design pertains to questions of product or system architecture, configuration, and layout; and as such, provides an important bridge between conceptual engineering design work and detailed design decisions. Prior research indicates that this phase of design is important to successful outcomes of design processes, yet it is not well understood. This paper reports the development and experimental testing of a system-level design tool that aids student designers in improving design quality through enhanced understanding of system level design issues and alternative selection. We found that use of the tool produced significant improvements in design quality among two experimental groups of mechanical engineering students. By developing and testing this tool, we hope to prove the usefulness of tools that aid engineering design at a system level, particularly in reducing heavily iterative and nonproductive tasks.

Introduction

Much knowledge about design process is gained directly from experience. As engineers become more practiced at design, their outcomes tend to improve, and presumably, so do their processes. Many methods and tools have been developed to teach good practices or otherwise assist in increasing the effectiveness of the design process. But two problems exist with the methods and tools currently available. First, few of the tools and methods have been verified experimentally. For design methodology to mature as a research field, greater rigor is needed in support of design methods and tools that purportedly improve the design process. Secondly, existing tools and methods focus on the "book ends" of the design process: numerous tools and techniques are available for the front-end tasks of conceptual design, particularly ideation and selection, and many tools and methods are available to assist with detailed design phases on the back-end, such as the wide-spread availability of computer-aided design and analysis tools. However, very little is available for the phase that bridges concept and detailed design, a phase we term system-level design (SLD).

In prior research on student design projects, we have observed that SLD effort associates positively with productivity and design quality. These indications have been observed through statistical analyses but have not been independently verified through experimentation. We, therefore, aim to better understand the impact of system-level design experimentally. To do this, we developed a tool designed to illicit system-level design work from the user. We then designed an experiment to test whether use of the tool improves design performance among student designers. This paper describes the experimental design and reports on the results obtained.

This work contributes to the field on a number fronts. First, the experimental approach used can serve as model for other researchers wanting to validate the efficacy of a design method or tool. Second, the experiment provides strong evidence that system-level design is, indeed, a high leverage opportunity for design teams. Simply put, even a small amount of design effort targeted at system-level issues can significantly improve design outcomes. And thirdly, through design and execution of the experiment, we have developed a deeper understanding of the important characteristics the concept-to-detailed design bridge.

The outline for the paper is as follows: The ensuing section describes background information for the study, including a definition of system-level design, brief literature review, and summary of prior work. Next, we describe the tool developed to aid with system-level design, followed by the experimental design. We then present and discuss the results, and conclude with the key learnings from this study.

Background

The first design phase following need identification is generally concept design; that is, addressing a given problem with preliminary ideas, strategies and approaches. Pahl and Beitz [1] define conceptual design as "the identification of the essential problems through abstraction … the basic solution path is laid down through the elaboration of a solution principle" (pg 139). This first stage of the design process includes problem definition, information gathering, and idea generation activities. The purpose of this stage is to clearly map the requirements of the problem and then generate possible solution approaches for meeting those requirements. The final step of conceptual level design is usually the selection of which solution method will be pursued.

Detailed design occurs when a designer is quantifying specific features needed to realize a particular concept. According to Pahl and Beitz [1], detail design "completes the embodiment of technical products with final instructions about the layout, forms, dimensions, and surface properties of all individual components" (pg. 400).

An important set of design decisions lie between conceptual level design and detailed level design. This area, which we call system-level design (SLD), is defined as *exploration of and decisions about components and subsystems, and their configuration*. This includes functionality, location, orientation, grouping, interfacing, and connectivity in relation to other components/subsystems and to the operating environment. These activities often relate to decisions about product architecture, product planning, and defining an assembly scheme. SLD starts with the solution path laid out in the conceptual design and expands the particulars of that concept. For each particular, whether it is a subsystem, a component, or an interface, options and alternatives can be researched and developed. Any alternative chosen must meet the criteria of the overall problem. These criteria might be "hard" engineering requirements, such as space or power, but could also fall into "softer" areas of concern such as economic or ethical requirements. The evaluation and selection of subsystems is extremely important to the overall success of a design project, and it occurs in SLD.

Pahl and Beitz [1] have posited rules and guidelines to embodiment or system design. These rules and guidelines involve "a flexible approach with many iterations and changes of focus." Such lack of specificity in the application of system design contrasts strongly with the more elaborate procedures that typify design process at conceptual and detailed levels. This lack of specificity for system level design seems endemic of design methods that include an intermediate stage between concept and detail design. This can perhaps be attributed to the many interdependencies of system level issues that make rapid convergence of the solution space difficult and highly iterative. We believe that SLD issues carry great importance in the design process, and do not have to be laden with iteration or changes of focus.

These issues are of increasing importance due to the growing complexity of designs. With greater complexity the linkage between concept design and development and detailed design becomes murkier. While certain lessons or rules may apply to particular cases they do not always produce optimal or even feasible results when applied outside their bailiwick. Bridging concept and detailed design for a complex set of needs requires the same kind of generally applicable methodology that we see at concept and detail levels.

Interestingly, system level design has not been as exhaustively studied. Generally, what information that is available are specific experiences of a designer or researcher that is applicable only to a few designs. While this understanding has been developed over many years and contains many valuable insights, it remains piecemeal. These designers and researchers often state the importance of system level design, based on their experience, but do not supply a method or tool to fill that gap. One of the difficulties in understanding system level design is that it does not lend itself to the prescribed standards and methods that work for conceptual and detailed design. Since our definition of system level design encompasses elements from embodiment design, system architecture, preliminary design, and modularity [1, 2, 3, 4], evaluation at a system level can require balancing complex and often competing objectives.

Relevant Literature

Verma has found that design experience and understanding of design theory are related but that the type of design experience doesn't make a statistical difference [5]. Design experience and judgment can only be built over time [6]. Many student designers do not have the reservoir of experience or judgment to draw upon that professional designers have. Thus, it is often very useful to support the novice designer's decision making process with a tool or method. To date these tools and methods have been developed based on the experience of authors rather than any planned or validated research [7, 8].

Conceptual design, including ideation and selection, is often considered the most important area of decision making in the design process, and many methods and tools have been developed to support it. However, none of the popular concept selection methods adequately support decisions that couple boundaries or provide the interface couplings between systems [9]. These interface couplings often result in trade-offs that are difficult to evaluate at a conceptual level [10]; however, they can often be evaluated effectively at a system level.

Addressing problems at a systems level requires a similar development challenge as at a conceptual level [11]. One of the most important strategic reasons for utilizing system level design is to provide a structured approach to dealing with complexity [12]. At a systems level, functions' interactions, functional schemata, and function criteria determine the physical "chunks" or subsystems that make up the product. Each of these subsystems will in turn be constructed of a physical architecture and physical interactions. Each of these interactions and interfaces will have to be addressed to complete the design process. But are they being addressed in a consistent and systematic manner?

Flemming, et al observed a tremendous loss of efficiency in detailed design due to small adjustments of existing design dimensions [13]. Sosa, Eppinger, and Rowles have identified 7 types of interfaces that are not typically addressed explicitly in design team work [2]. Also, consensus in identifying modules is rarely reached in team design work [3]. These difficulties emphasize the need to for a systematic method of identifying subsystems and then addressing solutions for them. Many unexpected difficulties that design teams encounter stem from unaddressed interface issues.

One of the methods used for systematic identification of systems is function-based design [14]. This design method is popular because of its application to computer based support. It allows for several aggregation techniques—the identification of modules, input-output flows, internal flows, materials flows, and product-specification functionality—which makes the method applicable to a wide range of problems. Once the aggregation has been determined, a formal functional decomposition can be used to describe what the product must do. The final goal of this method is "a clearer understanding of the set of related functions necessary to allow the product to fulfill its overall intended function (pg 414)" [15]. Identifying the systems of a design problem is only the first difficulty to overcome. Once the product is aggregated systematically it is necessary to develop a rule set to determine when and where to apply system level design methods [16]. In order to create a rule base for a system level design tool we will look to our own experience with system level design.

Prior work

Our research group identified a statistical correlation between SLD documented in student design journals and project outcomes in analyses to isolate design process elements that contribute to good designs [17, 18, 19, 20]. A number of themes emerged from this work, but perhaps the strongest was that design activity that occurs at a system level correlates highly to both design team productivity and outcome quality. This research indicated that a possible causal relationship could be shown to exist if we could develop a test to isolate the effects of SLD on design process. An initial screening experiment [21] was able to strengthen the correlation in an experimental environment but was not able to establish a causal relationship.

This screening experiment brought to light many difficulties in designing an experiment to test a specific aspect of the design process. From these learnings, we adapted a tool to test very specific aspects of system level design that we felt would benefit a general design process; namely, we focused our tool on the design of interfaces between functional subsystems. Based upon our prior work, we felt that the application of our tool prior to committing to a final conceptual design and before beginning detailed level design would result in better designs.

Morphological Tool Adaptation

Morphological matrix tools are often used to aid in ideation using a systematic method of developing and combining potential design solutions [6, 12]. These tools prompt the designer to identify the sub-functions needed to meet the stated design requirements, and then to brainstorm different ways the sub-functions can be accomplished. The combination of sub-function alternatives generates a large number of concept alternatives for the overall concept. However, while this tool provides possible working structures, it doesn't explicitly address system level design issues. Identifying the components and the structure addresses only part of the necessary design issues. Interface configuration, orientation, grouping, and connectivity (user and environment) all play vital roles in the success of a design but are not normally considered explicitly in the typical morphological analysis, even though these issues all contribute to the feasibility and desirability of a concept.

We theorize that addressing system-level issues *before* making concept selection decisions will greatly enhance the quality of such decisions and avoid difficulties later in the project due to flawed design concepts. One very important SLD issue is the interface configuration between functional components. Interfaces are increasingly important as the complexity of the design problem grows but interfaces are rarely considered as part of the overall design. By addressing the key interface configuration issues using a morphological tool, we feel that SLD can be leveraged to improve the outcomes of engineering design efforts.

Tool Description

The adaptation of the general morphological tool requires that designers have identified alternative conceptual designs and narrowed the set to a handful of promising alternatives. Each conceptual design idea is then analyzed at a system level. This analysis begins by identifying the key functions that the concept must execute to achieve the overall design objective. Each of the functions for a given concept may have multiple implementation options, each having different interface requirements (see Figure 1). These interface requirements may create many unique interface configuration options when combined with the other functional options. Once the morphological matrix has been filled in with the functional options, functional incompatibilities (exclusions) and necessities (dependencies) are identified. From there, the designer generates a list of potentially feasible alternative configurations for this particular concept. One way to generate alternative configurations is to create combinations of options that optimize each of the different functions.

CONCEPT: ____________________________

	Option A	Option B	Option C
Function 1	1A	1B	1C
Function 2	2A	2B	2C
Function 3	3A	3B	3C
Function 4	4A	4B	4C

Dependencies:

Exclusions:

Feasible Combinations:

FIGURE 1: MORPHOLOGICAL SYSTEM DESIGN TOOL (MSDT) TEMPLATE

Interfaces discussion

Once the list of potential combinations has been identified, a more informed discussion of the interfaces can be made. This discussion expands from the given interfaces between the functional components and begins by identifying additional interfaces. These additional interfaces might be with a user, with the environment, or with another device. These broader scope interfaces serve to set the stage for a story-telling walk-through of the design. This walk-through focuses on describing the functional path of the design objectives in terms of the interface requirements. At each point of this walk-through the designer considers how the interfaces are handled and whether alternative methods might exist for meeting the interface requirements.

It is important to address the interface issues at a non-superficial level. For example, if an exclusion or dependency exists between two options, then the constraints that create the exclusion or dependency should be identified. The answers to these questions often lead to potential revisions to interface configurations. This discussion relies heavily on designer expertise because it is based upon the expected behavior of the system; however, the morphological tool serves as a boundary object against which expectations can be checked.

Feasibility check

The final step before proceeding to concept selection is a reality check: is the concept, as configured, feasible? Can it be made feasible? Or is more information needed to make the determination? This check is made against the overall design specifications, not merely against subsystem requirements. For simple problems this can often be answered by inspection but more complex problems may require some modeling. Since the answers to these questions may not be definitive, the feasibility check can be used to identify areas that require additional background research.

These steps are then repeated for each concept alternative under consideration. The result of this method is a set of best configurations for each concept. This means that during concept selection, rather than comparing the preconceived versions of concepts, designers compare the best known configurations for the concept alternatives. Additionally, the concept selection decision inherently considers subsystem interface issues. An overview of the process to incorporate the morphological system-level design tool (MSDT) is displayed in Figure 2.

1. Generate conceptual design alternatives
   a. Define the problem
   b. Generate alternatives
   c. Narrow the set of alternatives to a manageable size

2. Apply MSDT to one alternative
   a. Identify the key functions of the concept needed to realize the design objectives
   b. Generate 2 or more options to accomplish each function
   c. Identify which function options cannot be used with other options (exclusions)
   d. Identify which function options require inclusion of another option (dependencies)
   e. Generate a list of alternative configurations for the concept alternative
   f. Investigate interface feasibilities
   g. Select the most promising configuration

3. Repeat step 2 for each alternative

4. Compare alternatives using the best configurations

5. Choose the best alternative

FIGURE 2: OVERVIEW OF MSDT PROCESS

Statement of Hypothesis:

Given that prior research results on student journal data pointed toward system-level design as a potential leverage point for improved design performance, we suspect that better designs will result from processes that use the morphological system-level design tool and corresponding system interfaces discussion. Or, more formally:

> <u>Hypothesis</u>: *Design processes which use the morphological system-level design tool followed by a discussion of system interfaces will produce better designs than design processes which do not.*

Since the morphological tool and method were designed to illicit system-level design considerations, this hypothesis implies that design processes which systematically consider system-level issues will outperform processes which do not. The remainder of this section details an experiment to test the stated hypothesis among senior mechanical engineering students.

Experimental Design

The experiment was designed as a crossover design, as depicted in Figure 3. A crossover design is a special type of repeated measurement experiment where experimental units are given different treatments over time, and compare pretest data to posttest data. In a crossover design each experimental unit serves as its own control. However, certain pitfalls must be avoided to ensure significant results. One such pitfall is the treatment of randomized experimental units. We randomized the assignment of teams to the two groups of the experimental protocol. The comparisons of primary interest are scores of the two problems between runs. Secondarily, we are interested in changes in performance between runs of the two groups, if comparability between problems can be established.

The arrows in Figure 3 show the expected comparisons and the directions of the improvement we hypothesize. Note that no difference is expected to exist between the golf ball and mouse problems in run 1 or run 2. This is important because the presence of a difference would mean that we do not have comparability is between design problems.

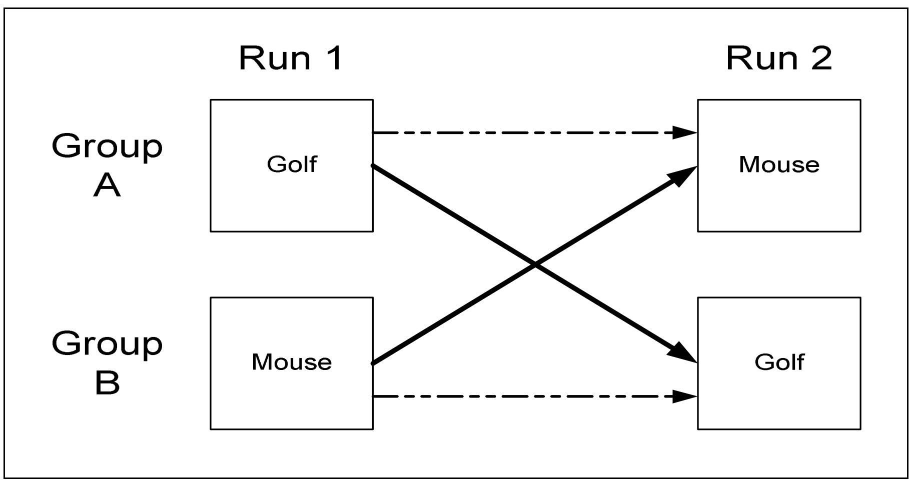

FIGURE 3: GRAPHICAL DEPICTION OF CROSS-OVER DESIGN

This type of design eliminates the ethical question of exposing a group of students to a potentially beneficial treatment without giving that same treatment to the control group. The experimental design also has good external validity. The comparisons allow for clear results that are either positive or negative, with little room for grey areas that might cloud the results. This method allows for within group testing and randomization without the need for large sample sizes.

A disadvantage of this design, however, is that it is not a “true” experiment since we do not randomize the second run, making internal validity potentially less robust. In addition, bias may enter into the results due to participants learning or training between the experimental runs or other sequential effects. But these biases are can be somewhat balanced by the timing and implementation of the design. For example time was extended between runs to lower the chance of sequential learning effects. Internal validity tests were added to the comparisons in order to better understand dependencies, and extra care was made to limit the amount of design process learning the teams experienced between runs.

Participants

The participants were mechanical engineering students enrolled in ME 403 Mechanical Engineering Design I. This course is structured as a design project experience emphasizing use of a formal design process, presentations, and documentation. The course also includes coverage of industry machining and welding practices. Each team was comprised of 2 members. In the final analysis there were 7 teams in group A and 7 teams in group B.

TABLE 1: ME 403 PARTICIPANT DEMOGRAPHICS

	Number of Participants	Average Age	Average GPA
Male	30	22	3.09
Female	2	21	3.25
Cumulative	32	22	3.10

Design problems

The problems used in the experiment are simple and straightforward (i.e., easy enough to be solved and implemented within a two-hour time window), yet designed to have competing design objectives and to be complex enough not to have obvious solutions. The first problem is that of moving a golf ball to a target while negotiating a drop of 10 inches. The problem statement is as follows.

Move a golf ball from a stand still in the starting area so that it comes to rest on the target ring as close to center as possible using only the materials provided. The only energy that can be applied to the ball must be stored in the materials.
Points will be awarded based upon the final resting location of the golf ball in relation to the target area. The objective is to score the most points possible while using a minimum number of parts.

The measurable quantities from this problem are the location of the golf ball when it comes to rest and the number of parts used to make the device. Since the target area for this problem is a set of concentric rings, the final resting location of the ball was determined by where the ball physically touched the target. If the design precluded the golf ball touching the target then the location of the ball was visually projected down onto the target to determine its location.

The second problem was to transport a hacky-sack to a linear target area. The primary difficulty of this problem was that the target area was strongly defined on one side and weakly on the other. The problem statement is:

Move the mouse (hacky-sack) from the starting line to a distance of no less than 3' and no more than 4'. Within that distance specification a point gradient exists from a maximum of 100 points at 3' to 25 points at 4'. Outside of this specification window no points are rewarded.
Points will be awarded based upon a combination of the final location of the mouse within the specification window and the number of parts used in the design. The objective is to score the most points possible in three runs while using a minimum number of parts.

The measurable quantities from this problem are the location of the hacky-sack when it comes to rest and the number of parts used to make the device. Since the hacky-sack was easily deformable, the location of the hacky-sack was determined by projecting the cross-section of the target area through the hacky-sack.

Both of these problems could only be solved using a set of parts that included an assortment of Lego parts, string, and a rubber band. The only difference between the problems was the amount of string (24 inches for the golf ball problem and 48 inches for the hacky-sack problem) and the number of wheels supplied (4 wheels for the golf ball problem and 6 wheels for the hacky-sack problem). Both problems featured the dual objectives of the device's accuracy and using a minimum number of parts.

Experimental Protocol

During Run 1, a brief familiarization exercise preceded the actual design problem in order to introduce students to the properties and capabilities of the materials that will be used during the exercise. After completing the familiarization, the experimenter presented a written problem statement to the group. After the group read through the design problem, the experimenter walked through the course that the prototype will have to navigate. During this period the groups were encouraged to ask questions about the problem statement or the course. The experimenter also reviewed the protocol that was to be followed.

The next step in the protocol was to generate at least three ideas, and select the best idea to prototype. Groups were required to produce and turn in sketches of the three most promising ideas, indication of the selected alternative, and their selection criteria at the end of 30 minutes. During this period, the design participants could pick up and handle the materials, but were not to assemble substructures to test ideas.

Once the design documentation was complete, the participants built prototypes of their best idea using the materials provided, and could test their designs on the test course. After a maximum of 20 minutes, the participants demonstrated their prototype in 3 consecutive test runs, with scores duly noted. Calibration of the prototype was allowed between trials as long as no changes were made to the prototype.

Run 2 followed the same basic design process as Run 1, except for the introduction of the MSDT protocol previously described. The MSDT was conducted after participants generated a number of conceptual design ideas, but before they narrowed to a single alternative. The idea generation and prototype building stages were kept identical in order to isolate the effects of the system-level design tool. Since the student participants were only superficially familiar with the MSDT protocol (was introduced in lecture the week before), the experimenter walked the participants through the sequence of steps, trained the students in the tool, and guided them in applying the tool to their problem. The experimenter was careful not to suggest design ideas or identify potential problems, but merely asked the participants to execute each step as indicated in the protocol (see Figure 2).

The overall amount of time allocated to execute a run had been used in previous experiments. The time allocated allowed most teams to finish the design problem fairly comfortably within the timeframe; some groups finished a little early while some had to push to finish, but all groups completed a testable prototype.

Analysis and Results

The two measurable quantities from the experiment were the performance score of the team's prototype over three trial runs, and the number of Lego pieces used in the prototype. The two factors were normalized and combined in order to obtain a single measure of design quality for each participant team. For the test scores, 300 was the best possible outcome so the scores were normalized against 300. For the piece count score, the least number of pieces possible was 1 so it was normalized against 1. The two normalized scores were then averaged, creating a range of scores from 0 to 1 with 1 being the best possible combined score.

When checking the normality assumption of the data using a normal probability plot, two potential outliers were identified. No assignable cause was found for the first data point; however the second data point traced back to a group that completely failed to follow experimental protocol during the second run. This group resisted the use of the morphological tool and after the interface discussion rejected not only the results based upon the tool but also the results of their initial conceptual level design work. They spent the prototype period "tinkering" with the parts and produced a poorly performing prototype. This data point was classified as an outlier due to failure to follow protocol, and was removed from the sample.

Test of Variances

The first step in the analysis of the normalized response variable was to test for equal variance between the key categories of comparison. This test determines which means test must be used and helps identify the appropriateness of the experimental design. Across all categories of comparison, the variance tested as statistically equal according to two-sample F-tests for variances. This test relies on the normality assumption noted above.

Tests of Means

Following the test of variance, a test of means was conducted using a two-sample student t-test assuming equal means. Each category was tested to determine whether the run 2 results were higher than the run 1 results, thus a positive difference indicates support for the hypothesis. Table 2 displays the means tests results.

TABLE 2: RESULTS OF STUDENT T-TEST OF MEANS ASSUMING EQUAL VARIANCE

	Run 1	Run 2	Difference
Golf ball problem	0.584	0.853	0.269**
Mouse problem	0.433	0.740	0.307**
Within-run difference	0.152	0.113	
Group A	0.584	0.740	0.156
Group B	0.433	0.853	0.421**

* p-value ≤ 0.10, ** p-value ≤ 0.05

Comparison of problems across runs

The first comparison of interest is whether any improvement occurred from one run to the next on the same problem. The golf ball problem displayed the expected improvement between the first run and the second run. The scores increased by a normalized value of 0.269 between Run 1 and Run 2, a 46% increase. The statistical significance of this improvement was very strong having a p-value of 0.028. The mouse transport problem displayed a similar improvement between runs. An increase in the normalized score of 0.307, or 71% improvement, was observed. This was also highly significant with a p-value of 0.027. These results strongly support the stated hypothesis.

Comparison of groups across runs

Next, the comparison of groups across runs compared the run 1 scores against the run 2 scores of groups A and B. In both cases, improvement in normalized scores was shown; however, the improvement of group A was not found to be statistically significant despite being an improvement of 26% over of the first run normalized score. The p-value for the group A test was 0.142 which fell outside the boundaries for significance established for this experiment. Group B showed a 97% increase in normalized score that tested significant with a p-value of 0.004. This highly significant result supports the hypothesis that use of the SLD tool improves design performance.

Comparison of runs

The final comparison of interest was a check that the individual runs were comparable. For this test we would expect no significant difference between the groups since the experimental groups were randomly assigned and the problems were of a similar level of difficulty. The comparison of the first run showed that the mouse problem was slightly easier with an average difference of 0.152 in normalized score. The p-value for this comparison was 0.331. The second run again showed that the mouse problem was slightly easier with a normalized score difference of 0.113. This failed to test significant at a p-value of 0.366. These tests indicate that there is not a statistical difference in difficulty between the two problems used, although it appears that the mouse problem may be slightly less difficult.

Feedback Survey

A post-experiment survey of the students was administered before the initial results of the experiment were reported to them. This survey asked the students to evaluate their experimental experience as related to their educational experience. From this survey, 89% of the participants felt that the design experience was worthwhile. 59% felt that the SLD tool was moderately to very helpful. 41% of the students could envision themselves using this tool on a future design project while 59% felt that they would be willing to use the tool on a future project. While not conclusive, the survey results triangulate with the experimental results indicating that the observed performance improvement was due to use of the SLD tool.

Discussion

The variance and means tests conducted on the normalized scores of the experimental groups result in a set of comparisons that support the hypothesis of the experiment. The non-significant result of the means test between runs for group A does not supply contradicting evidence against the hypothesis. The other three comparisons are highly significant and support the stated hypothesis of the paper. Figure 4 displays these results graphically.

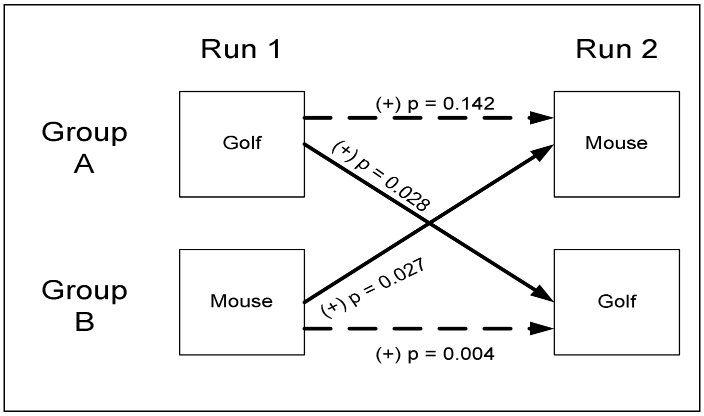

FIGURE 4: GRAPHICAL SUMMARY OF THE COMPARISON RESULTS

The primary comparison of interest to us is performance between the problems: will groups using the SLD tool have a better performance on a given problem than groups that do not use the tool? These two comparisons are represented graphically in Figure 4 by the crossing arrows. A fundamental assumption for this comparison is the comparability of the sample populations. This assumption is allowable since the two experimental groups were pulled from the same population and randomly assigned to the experimental groups for the first run. The results of this comparison are strongly in support of the stated hypothesis. The p-values are strong indications of significance, and the relative improvement between runs is fairly large (46% and 71%) due to the introduction of the MSDT.

The second comparison is that of the experimental groups before and after the introduction of the MSDT. According to our hypothesis, the groups should show an improvement on a problem when using the MSDT as opposed to when they do not use the tool. This comparison spans the two problems used in the laboratory and therefore rely on the comparability of the two problems. This comparability was tested in the between runs means and variance tests. In both cases the tests showed no statistically significant difference between the experimental problems. Therefore the comparison of groups is allowable.

The test comparing the performance of group A did not show a statistically significant improvement. The p-value of this test was 0.143 which fell outside of the level of significance set for this experiment (LOS = 0.10). However despite not being statistically significant, there was an improvement in the scores seen for group A from run 1 to run 2. This result does not provide strong support for the hypothesis but it does not contradict it either. Group B on the other hand showed a very high level of significance. The p-value for this test was 0.004. This significant improvement for group B strongly supported the stated hypothesis.

Several assumptions made in this experiment concern comparability. The most important assumption is that the two groups represent samples of the population that are similar enough to make comparisons against. In order to assure this is the case, members were randomly assigned to the experimental groups. The second assumption of comparability is that between the problems. Pre-screening of the problems indicated that they were of similar difficulty but until we performed the test of means between the runs we couldn’t be sure our assumption was justified. It turns out that the problems showed no statistically significant differences.

An area of future work is the application of this method to problems of greater complexity. The problems used in this experiment were simple and straightforward. A best solution existed and could be realized in 1-2 hours by fairly novice design engineers. The MSDT was successful at identifying the superior designs, which were not readily apparent from the start, for these simple problems. We suspect that the tool's usefulness in discovering better designs will actually increase with the complexity of the design problems since interface issues become more critical with increasing complexity; but this remains to be tested.

Tandem to this question is the applicability of the method to other design domains. Is the method also useful for other areas of engineering design, such as electrical engineering? To effectively address this question requires a more fundamental understanding of why the tool is beneficial. For example, is systematic exploration of subsystem alternatives, or the discussion of interfaces that is the key to better outcomes? Or, perhaps, one needs both; either one on its own has little value. Investigation into the potential cause of the tool's benefit would be needed to gain this more fundamental insight. The answers obtained here would likely raise the further question of whether other SLD tools could or should be created to aid in the exploration and improvement of designs at a systems level.

The experiment is not without potential biases, the most serious of which may that participants learned something in the first run that helped them improve performance in the second. We took two actions to counter this potential bias. The first was to include a materials familiarization exercise at the beginning of the first run. This exercise introduced the students to the parts and general structures before either run to minimize the effects of materials familiarity. The second precaution was to delay the second run of the experiment for approximately six weeks. This delay made it difficult to reconstruct the details of their previous designs. The hazard associated with this delay is that the students might learn from an outside source or their own experience; however there was no indication in the observations made during the experiment that such learning occurred. For example, the students' prototyping skills did not appear appreciably greater in the second run than immediately after the initial familiarization exercise. Also, the amount and nature of concept-level idea generation was comparable between groups and between runs.

A final source of possible bias is the interaction of the experimenter with the participants. In order to guide the students in use of the novel design tool, the experimenter necessarily interacted with the participants more in Run 2 than in Run 1. It is possible that simply interaction with an "expert" lead to superior results irrespective of the content of that interaction. However, the post-mortem survey provides additional evidence to say that the tool, and the system-level design issues it brought to the surface, was useful and explains the results.

Conclusions

The evidence presented here seems to provide clear indication that the MSDT has a strong impact on the outcome of design. Improvement was seen across every comparison that was predicated in the hypothesis, and in 3 of the 4 cases, the improvement was highly significant. This supports the conclusion that system level design is important to design success, but more immediately suggests that the tool proposed is a successful instrument to leverage design outcome.

The evidence suggests that use of this tool improves the outcome of the design process. We believe that the reason for the improvement is that the students using the tool came to a deeper understanding of the system level design issues that typically might not be addressed in the design selection process only to be discovered late in the process (e.g., in prototype building or testing). From our observations of the participants, this deeper understand made the selection of concept easier and less mysterious. This aided in the transition between concept and detailed design by giving the groups more confidence in their decisions and decreasing the amount of "tinkering" in the detail design phase. In all, the students produced higher quality designs by emphasizing the SLD issues contained in the MSDT.

Perhaps just as importantly, this study provides a template for demonstrating the usefulness of a design tool or method. To date, few of the design tools or methods advocated in the design theory literature have been validated empirically. Outlined in this paper is one approach for doing so that has a fairly straightforward protocol and simple diagnostics, yet contains statistically rigorous internal and external validation. This work will hopefully aid in the validation of additional tools and methods.

References

[1] Pahl, Gerhard & Beitz, Wolfgang, 1996, *Engineering Design: A Systematic Approach*, Springer, London, pg. 139, 400.

[2] Sosa, Manuel E., Eppinger, Steven D., & Rowles, Craig M., 2003, "Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions," *Journal of Mechanical Design*, Vol 125: pp 240-252.

[3] Gershenson, John K., Prasad, G. Jagganath, & Zhang, Ying, 2004, "Product Modularity: Measures and Design Methods," *Journal of Engineering Design*, Vol 15, No. 1: pp 33-51.

[4] Van Wie, Mike J., Rajan, Palani, Campbell, Mattfew I., Stone, Robert B., & Wood, Kristin L., 2003, "Representing Product Architecture," ASME 2003 Design Engineering Technical Conference, Chicago, Illinois, USA.

[5] Verma, Niraj, 1997, "Design Theory Education: How Useful is Previous Design Experience?," *Design Studies*, Vol 18: pp 89-99.

[6] Holt, J.E., 1997, "The Designer's Judgement," *Design Studies*, Vol 18: pp 113-123.

[7] Wallace, Ken, & Burgess, Stuart, 1995, "Methods and Tools for Decision Making in Engineering Design," *Design Studies*, Vol 16: pp 429-446.

[8] Committee on Engineering Design Theory and Methodology, 1991, "Improving Engineering Design," National Research Council.

[9] King, Andrew M. & Sivaloganathan, Siva, 1999, "Development of a Methodology for Concept Selection in Flexible Design Studies," *Journal of Engineering Design*, Vol 10, No 4: pp 329-349.

[10] Ulrich, Karl, 1995, "The Role of Product Architecture in the Manufacturing Firm," *Research Policy*, Vol 24: pp 419-440.

[11] Ulrich, Karl T., & Eppinger, Steven D., 2000, *Product Design and Development*, 2nd ed., McGraw Hill, New York.

[12] Marshall, Russel & Leaney, Paul G., 2002, "Holonic Product Design: A Process for Modular Product Realization" *Journal of Engineering Design*, Vol 13, No. 4: pp 293-303.

[13] Flemming, Ulrich, Bhavnani, Suresh K., & John, Bonnie E., 1997 "Mismatched Metaphor: User vs. System Model in Computer-aided Drafting," *Design Studies*, Vol 18: pp 349-368.

[14] Verma, Manish & Wood, William H., 2003, "Functional Modeling: Toward a Common Language for Design and Reverse Engineering," ASME 2003 Design Engineering Technical Conference, Chicago, Illinois, USA.

[15] Dahmus, Jeffrey B., Gonzalez-Zugasti, Javier, & Otto, Kevin N., 2001, "Modular Product Architecture," *Design Studies*, Vol 22: pp 409-424.

[16] Sridharan, Prasanna & Campbell, Matthew I., 2004, "A Grammar for Function Structures," ASME 2004 Design Engineering Technical Conference, Salt Lake City, USA.

[17] Jain, Vikas K. & Sobek, II, Durward K., "Linking Design Process to Customer Satisfaction Through Virtual Design of Experiments," under review at *Research in Engineering Design*.

[18] Sobek, II, Durward K. & Jain, Vikas K., "Process Factors Affecting Design Quality: A Virtual Design of Experiments Approach," under review at *Journal of Mechanical Design*

[19] Costa, Ramon, & Sobek II, Durward K., 2004, "*How Process Affects Performance,*" ASME 2004 Design Engineering Technical Conference, Salt Lake City, USA.

[20] Wilkening, Samuel, & Sobek II, Durward K., 2004, "Relating Design Activity to Quality of Outcome," ASME 2004 Design Engineering Technical Conference, Salt Lake City, USA

[21] Ruder, Joshua A., & Sobek II, Durward K., 2005, "Student System Design Activities: An Empirical Pilot Study on Improving Design Outcomes," ASME 2005 Design Engineering Technical Conference, Long Beach, California, USA