Role of Axiomatic Design in Teaching Capstone Courses

**Edwin Odom, Steven Beyerlein, Christopher A. Brown,
Daniel Drew, Lloyd Gallup, Sam Zimmerman, and Jeremy Olberding**

University of Idaho/Worcester Polytechnic Institute

Abstract

Helping undergraduate engineering students learn effective design practices that are applicable to the modern workplace is one of the most complex challenges of engineering education. One strategy to help students master open-ended design projects is to use a systematic process. However, students often want to jump past the front end of the design process and this compromises the quality of the final product. This paper examines the suitability of Axiomatic Design in addressing this problem. Central to Axiomatic Design is early identification of uncoupled design parameters that address independent functional requirements. A new design process, incorporating Axiomatic Design methods along with the use of Acclaro software (http://www.axiomaticdesign.com) was developed in this work and piloted with several capstone design teams at the University of Idaho during the current academic year. Early indications are that these teams were more successful in establishing functional requirements that were more complete, more logically hierarchical, and more independent than other design teams. Furthermore, design ideas discussed by these teams seemed to be accepted or rejected on their own technical merits, rather than the force of the personalities of students who presented them. Thus, we have concluded that axiomatic design helps capstone teams produce higher quality design projects.

1. Introduction

This paper explores benefits and issues associated with incorporating Axiomatic Design into a capstone design experience. Specifically, we wanted to determine if Axiomatic Design could alleviate difficulties encountered when using a traditional algorithmic approach to design. We focused on the problem definition, solidification of the conceptual design, and the completion of CAD drawings that identify design parameters that satisfy functional requirements.

In the traditional algorithmic approach, which has been used with over one hundred capstone design projects at the University of Idaho, about half the projects suffer from the need to rectify unforeseen problems that necessitate substantial iteration and additional expenditures late in the design process. This approach involves problem definition with a list of "musts" and "shoulds," idea generation, and concept selection. The approach is algorithmic in that a sequence of activities is prescribed, and the implication is that a good design process results in a good design product. In this process, there are no generally applicable rules to assess the success of the design, other that how well the design complies with the list of "musts" and "shoulds."

Axiomatic Design, in contrast to the algorithmic design, is based on two axioms, and the contention is that all good designs comply with these two axioms[1]. The application of these two axioms supplies the basis for objective assessment of design solutions. Axiom one says that the best designs are the ones with the most independence of the functional elements. Lack of independence, or coupling, is the cause of lack of adjustability, or poor control. The result is that iterations are required to satisfy the functional requirements, i.e., "musts" and "shoulds," of the design. The second axiom is that among the independent design solutions the design with the least information content will be the best. Information in this sense is numerically equal to the log of the inverse of the probability of success, such that minimizing the information content is equivalent to maximizing the probability of success. Applying axiom one, the independence axiom, requires that the design have a hierarchical structure in two domains, the functional and physical. To apply axiom one, interaction matrices between the functional and physical domains are generated at each level and for each branch in the hierarchy. Alternative design solutions are examined based on how they influence coupling between different design parameters[1].

Like algorithmic design, Axiomatic Design is process-oriented. The development of the hierarchies of functional requirements--the elements of the functional domain--and design parameters--the elements of the physical domain--requires following a sequence of activities. It is, in part, the development of the sequence of activities that creates some of the issues associated with the implementation of axiomatic design, but ultimately this sequence contributes to its benefits as well. The published work on axiomatic design has concentrated on the application of the axioms to fully developed designs[1,2]. The literature is largely silent on the process of developing the functional and physical hierarchies, and on facilitating the use of axiomatic designs with groups of novice designers.

Axiomatic Design is attractive because of its criteria for evaluating designs, its standard format for recording design decisions, and its ability to promote team-based discussion about design requirements and associated design parameters. The case study approach we adopted to answer the research question that had three main aspects:

1. analysis of completion dates for different phases of the design process by previous senior design teams,
2. distilling key axiomatic design ideas in a toolkit that could benefit capstone design projects, and
3. piloting the toolkit with several senior design teams to see how this impacted behavior during conceptual design and system integration.

Through the pilot projects we wanted to further explore issues and benefits associated with the use of Axiomatic Design in capstone design projects, compared with the traditional, algorithmic design process[3]. We made extensive use of Acclaro, specialized software developed ADSI in Brighton, Massachusetts (www.axiomaticdesign.com) to facilitate the development of design hierarchies and of the design matrixes required for applying the independence axiom.

2. Analysis of Capstone Projects

The capstone design experience at the University of Idaho is a two-semester sequence that involves a variety of projects sponsored by regional industry. This culminates in the largest academic design show in the Pacific Northwest where design teams from departments of Mechanical Engineering, Electrical Engineering, Civil Engineering, Chemical Engineering, Biological Systems Engineering, Metallurgical Engineering, and Computer Science display their work to the public. Information about this annual event can be found at www.uidaho.edu/expo.

During the current academic year, the Mechanical and Electrical Engineering have joined forces in an interdisciplinary class that meets concurrently and has uniform project expectations. More than 80 seniors and a half-dozen graduate student mentors are part of this combined enterprise. Roughly 1/3 of the design teams have both ME and EE members, 1/3 have only ME members, and 1/3 have only EE members. All teams are required to respond to the needs of an external customer, maintain personal logbooks, prepare a problem statement with specifications, present their solution in various design reviews, fabricate a working prototype, write a design report, and document their design process on a course web page (http://seniordesign.engr.uidaho.edu).

The structure of this year's joint ME/EE class is outlined in Table 1. The course is divided into seven phases that include supporting activities and a milestone requiring instructor approval before the team can move on to the next phase. Expected duration and effort required for each phase are also estimated in Table 1. This course design parallels many other capstone programs across the country[4,5] and follows the methodology advocated by popular design textbooks[6,7,8,9,10]. Our local implementation has evolved over the last ten years to align better with our design infrastructure and ABET learning outcomes[11,12,13]. The recently expanded class format has underscored the need to measure design quality early in the design process and provide feedback to design teams about what technical issues should be given top priority.

Our experience monitoring the progress of more than 100 year-long capstone projects at the University of Idaho is that a traditional "problem definition, and idea generation, and concept selection" approach to design is ultimately successful in producing a viable prototype for only one-half of the projects. The other one-half of the projects require substantial rework and expenditures late in the design process to rectify unforeseen problems.

This situation is illustrated in Table 2 that tracks completion dates for each phase by all of the ME design teams in 2003-04. Seven of the twelve projects delivered their design product and supporting documentation before the last day of class. These teams did not experience major difficulties during detail design that caused them to return to the conceptual design stage. Even so, it was interesting that many of these teams published elaborate and thoughtful lists of 'musts' and 'shoulds' on their project web page, but then took another 4-6 weeks to fully rediscover these during their exploration of various design alternatives. Despite being on schedule, these teams spent considerable time researching unproductive solution paths rather than conducting deeper analysis and testing of design concepts ultimately selected for adoption.

Table 1. TEAM ACTIVITIES & TIME ANALYSIS FOR CAPSTONE DESIGN

PHASE 0: Project Scoping and Team Selection
Begins in summer and extends through the first month of class when design teams are assigned

PHASE 1: Problem Definition → 3 weeks (25-30 hrs/person)
ACTIVITIES: regularly scheduled times/places to meet/work
team goals/roles/responsibilities/rules
system for organizing team documents
on-site customer interview
literature review/technology research
MILESTONE: approved web page (problem statement, specs, and research areas)

PHASE 2: Conceptual Design → 3 weeks (35-45 hrs/person)
ACTIVITIES: generation of design alternatives
supporting calculations/experiments
method and timeframe for component and/or software selection
preliminary budget estimate
formal design review with customer/advisors/mentors/instructors
MILESTONE: approved conceptual design based on oral presentation

PHASE 3: System Integration → 2 weeks (25-30 hrs/person)
ACTIVITIES: resolution of design review issues
system level DFMEA
finalized budget and work plan
written design proposal
MILESTONE: approved design proposal

PHASE 4: Detail Design → 3 weeks (40-50 hours/person)
ACTIVITIES: purchase orders prepared and issued, assembly drawings, detailed parts list, component drawings, fabrication review/shop plan/schedule
MILESTONE: approved drawing package and manufacturing plan

PHASE 5: Manufacturing → 3 weeks (35-45 hours/person)
ACTIVITIES: special training on equipment, part acquisition, part creation in shop
MILESTONE: demonstration of working hardware and/or software

PHASE 6: Testing → 3 weeks (35-45 hours/person)
ACTIVITIES: performance data, data analysis, and design evaluation (DFMEA)
MILESTONE: approved outline for final report

PHASE 7: Delivery → 2 weeks (20-30 hours/person)
MILESTONE: approved final design report, hardware/software delivery plan, archival web page, and project CD (solid models, reports, web page, etc.)

Table 2. Completion Dates for Different Design Phases by ME Teams (2003-04)

Team	Problem Definition	Conceptual Design	System Integration	Detail Design	Manufacturing	Testing	Delivery
1	10/10	11/20	12/20	2/1	3/1	4/1	5/1
2	11/1	12/1	2/1	2/15	3/15	4/1	5/1
3	11/1	11/20	12/15	2/15	3/15	4/10	5/1
4	11/1	12/10	2/1	3/1	4/1	4/10	5/5
5	10/15	11/15	12/15	2/1	3/1	4/10	5/5
6	10/20	11/15	12/15	1/20	3/1	4/20	5/5
7	10/20	11/20	12/20	2/1	3/15	4/20	5/5
8	***11/1***	*****1/15***	***2/15***	***3/1***	***4/1***	***5/1***	***5/10***
9	***10/20***	*****1/15***	***2/15***	***3/25***	***4/25***	***5/1***	***5/10***
10	***10/15***	***11/1***	***12/15***	***2/1***	***4/1***	*****5/1***	***5/15***
11	***10/15***	*****1/15***	***2/25***	***4/1***	***5/1***	***6/15***	***7/1***
12	***11/15***	***12/15***	*****2/1***	***3/1***	***4/1***	***7/1***	***8/1***

Five projects (shown in bold) were completed after the last day of class. Three more projects achieved completion by the end of finals week. Unfortunately, two projects required rework extending into the summer term. This was necessary to minimally meet customer expectations and sustain goodwill that is critical for obtaining support for future capstone projects. Upholding professional commitments is also a good lesson for our students to take to the workplace. The phase where it was evident that these teams were behind the rest of the class are shown with double asterisks. A universal characteristic of all five projects was selection of a design alternative that was initially appealing but which had negatively interacting design parameters. In four of the five cases this was resolved by clarifying the original problem definition and pursuing a new solution path late in the design process.

Team 8 had an entrepreneurship project and concluded their initial solution was much too costly and prone to error after their mid-year design review. Team 9 had a fixture with over 50 moving parts whose motion needed to be carefully orchestrated. The team was not able to integrate their many ideas into a final solution until they built and tested a wood prototype late in January. Manufacturing to desired tolerances required considerably more effort than anticipated. Team 10 worked on an apparatus over 8 feet tall that could manipulate canisters weighing several hundred pounds. They needed to make last minute changes after they experienced an unanticipated motor failure. Team 11 had a design with tight tolerances that were compromised in manufacturing. These were necessary for proper operation of a digital camera that was part of the system. The team also chose to implement an automated motion control system rather than customer-approved manual operation. This posed many perplexing software issues. Team 12 had a mechanical design that was done early, but their prototype involved sophisticated software development and debugging.

3. Axiomatic Design Toolkit for Capstone Courses

Axiomatic Design theory has emerged from MIT as a tool for streamlining product development and managing complex engineering projects[3]. Suh's books on axiomatic design[1,2] are the seminal works for advancing the science of design using axioms. They contain all the essential elements of Axiomatic Design, however it can be difficult and time consuming to develop teaching and application methods from these texts alone. This section of our paper outlines only the elements of Axiomatic Design that we deemed appropriate and necessary for a capstone course context. This elemental approach is based on the combined experiences at the University of Idaho and at Worcester Polytechnic Institute (WPI), where Axiomatic Design has been taught and used in a variety of applications since 1990. The experience in the development of the teaching and application techniques at WPI over the past 15 years was important in improving the implementation of the group at U of I. Our goal was to use Axiomatic Design to:

- Separate functional requirements from the methods of solution;
- Build foundational knowledge about the design problem, setting the stage for collecting relevant information in a timely manner;
- Minimize troublesome overlap in functional requirements;
- Avoid solution lock-on by the team or by the customer; and
- Document problem definition for use in evaluating design quality on ongoing basis by the team, the instructor, and the customer

Our conceptualization of Axiomatic Design consists of two axioms, two structures for decomposing design elements, and a methodology for design decomposition and subsequent physical integration. The axioms and structures are summarized in Table 3.

Axioms are statements that are accepted as true and cannot be proven. They may be accepted based on their intrinsic merit. They may be established rules or principles or self-evident truths. Or, they may be postulated as the basis for an argument. The latter is the perspective that we have chosen. The design axioms were proposed by Suh[1] based on studies of good engineering designs in multiple disciplines. The proposition is that the two axioms are true for all good designs. We recognize that Suh's design axioms are not universally accepted by the engineering design community. In fact, we use the existence of this skepticism pedagogically, challenging the students to examine the axioms critically and look for violations. The discussions of the potential violations have been instructional. All the contentions that have been put forward have been resolved in favor of the axioms.

The first axiom stresses simplicity in design by minimizing the coupling of functional requirements (FRs) with each other, and of unintended coupling of design parameters (DPs) with FRs, allowing for finer adjustment and control functions without need for iteration. Compliance with axiom one also avoids unintended consequences. The second axiom stresses robustness and reliability by minimizing the amount of information required for extended operation or repair. Compliance with these two axioms provides a basis for evaluating the quality of designs and can lead to important design innovations. The ability to assess design quality through axioms in this manner is the principle element that distinguishes axiomatic design from other design approaches.

Table 3. Axioms and Structures of Axiomatic Design

AXIOM #1: Maximize independence of functional requirements

AXIOM #2: Maximize probability of success

STRUCTURE #1: Lateral Decomposition (CNs + Cs) → FRs → DPs
Customer Needs (CNs)
Constraints (Cs)
Functional Requirements (FRs)
Design Parameters (DPs)

STRUCTURE #2: Vertical Decomposition (hierarchy in FRs & DPs)

Lateral decomposition separates customer needs (CNs) from constraints (Cs) that should never be violated. Customer needs logically lead to functional requirements that should be consistent with the first axiom. Functional requirements, in turn, lead to design parameters that should be consistent with both axioms. Guidelines for separating FRs, DPs, and Cs are given in Table 4. Vertical decomposition separates the design into different layers of detail in a design hierarchy. Higher-level DPs set the stage for lower level FRs. The process for developing the hierarchy is known as zigzagging between FRs and DPs, and this recursive aspect of Axiomatic Design is one of the elements that distinguish it most from traditional design methodologies that move sequentially from problem definition, to idea generation, to concept selection.

Table 4. Guidelines for Identifying FRs, DPs, and Cs

	Functional Requirements (FRs)	**Design Parameters (DPs)**	**Constraints (Cs)**
Definition	Describes what the design should do	Describes what the design should look like	Describes to what limits the design must conform
Purpose	Satisfy customer needs	Satisfy FRs	Define boundaries
Word Usage	Start with verbs (imperative phrases inspiring action)	Start with nouns (declarative phrases prescribing solutions)	
Tolerancing	Functional tolerances	Physical tolerances e.g., x = +/-.05 mm	Have limits e.g., x < .5m in length
Uniqueness	Independent from other FRs	One selected to satisfy each FR	May be linked To several FRs
Relation to Solution	Requires a DP	Specifies a purchased component, drawing, or a process	Must not have a DP

Table 5 presents a method for applying the axioms and using the structures to establish a hierarchy of FRs and DPs. This was created to accompany the Acclaro software, which helps visualize lateral and vertical decomposition, facilitates application of axiom one, and promotes design communication between team members, mentors, and instructors. The starting point for using the method in Table 5 is a customer interview addressing key questions about the design problem, background reading suggested by the instructor, an orientation on axiomatic design theory, a recorder proficient in the use of Acclaro software (www.axiomaticdesign.com), and a block of at least 90 minutes for the design team, their mentor, and their lead instructor to work together. The outcome from this process should be a clear definition and prioritization of customer needs, a list of realistic constraints, functional requirements (complete, decoupled, measurable, and organized in hierarchy), and tentative design parameters that are vectors for further research and refinement.

Table 5. Method to Determine FRs & DPs

STEP 1 : Write top level functional requirements (FRs)
STEP 2: Select top level design parameters (DPs)
STEP 3: Check Axiom 1 (minimize coupling)
STEP 4: Check Axiom 2 (maximize chances of success)
STEP 5: Verify that constraints are not violated (Cs)
STEP 6: Go to the next level down (writing FRs and selecting DPs)
STEP 7: Go to step 3, repeating until the solution is obvious
STEP 8: Visualize solution by showing DPs on system diagram

The zigzagging process for creating the FR-DP hierarchy is shown in Figure 1. The FRs at the highest level are formulated first in a solution neutral environment. The corresponding DPs are then chosen and checked with their generating FRs against the axioms. Next, the FRs at the next level down are formulated within the constraints supplied by the DPs at the higher levels, then the corresponding DPs are chosen. In this way, zigzagging between the functional and physical domains, the FR-DP hierarchy is developed.

Frequently, especially at the highest levels, the DPs will be stated in a general or generic way, as systems for fulfilling the FRs. The components of the system and the specific nature of the physical system are often only evident as the lower levels are reached. At the lower levels the components of the systems and sub-systems may frequently be mechanisms or devices. The nature of the system, mechanism or device can be reveled in modifiers, such as: mechanical, pneumatic or electrical. Competing DPs can be tested based on their ability to satisfy the FRs and fulfill the axioms.

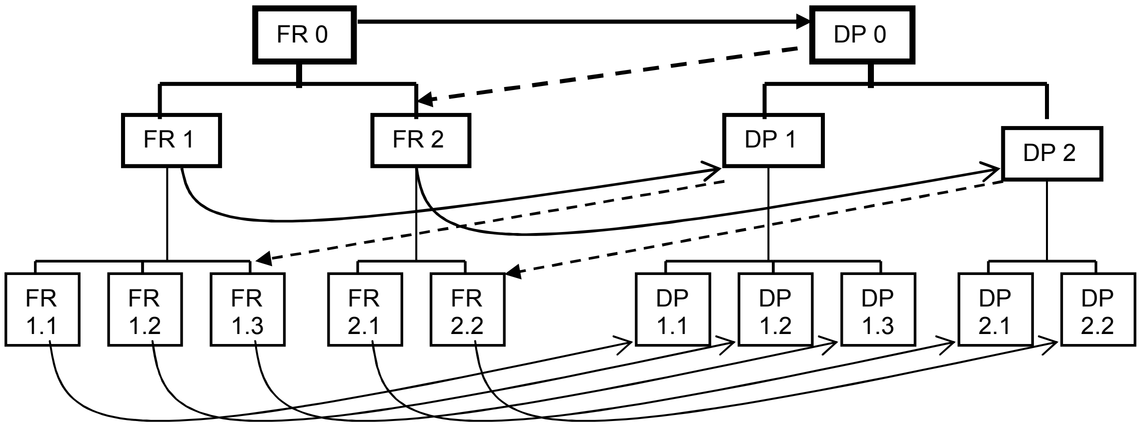

Figure 1. Zigzagging process for developing the FR-DP hierarchy.

The following tips are helpful in writing functional requirements (FRs):

- Top level FRs should be customer needs expressed in engineering language.
- Upper level FRs should capture all essential functionality, even if this must be stated in very general terms. In other words, lower level FRs should not introduce new requirements, although they may clarify or expand on existing requirements.
- Lower level FRs (children) should be derived by appropriately decomposing upper level FRs (parents). Note that children FRs may be different depending on the DP corresponding to the parent FR, since the DP explains how the parent FR will be met.
- Upper level FRs (parents) should either have no children or more than two children.
- At all levels, FRs should be (i) collectively exhaustive, (ii) mutually exclusive, (iii) equal in size/reducibility, (iv) a minimum set, and (v) bounded in magnitude.

The following tips are helpful in writing design parameters (DPs):

- DPs should focus on equipment/solutions.
- Select alternative DPs that are simplest to implement.
- If you have too many DPs, consider making some of them constants.

The following tips are helpful in checking Axiom 1:

- In examing interaction between DPs and FRs, ask yourself whether that DP could compromise each FR you have listed. If so, note this.
- Be wary of fundamental coupling due to relationships based on physics, chemistry, mechanics, or electronics. These are not easily resolved.
- During the decomposition phase do not be overly concerned about integration coupling resulting from physical connection of components, this should be considered during system integration rather than conceptual design.
- Use sequential coupling (relationships resulting from steps or stages in a process) to establish a logical order for addressing FRs and DPs.

4. Piloting the Toolkit: Reed Gouger Example

This section of the paper explores how the toolkit in the previous section can be used to advantage in a capstone design course. It centers on an ongoing project to replace a manually operated reed gouging device with a less labor-intensive device that affords more adjustability. Bassoon reeds are made from cane grown in the south of France. Finished reeds are commercially available. However, they cost about $20 each. Because a single reed will only be used between 3-8 weeks, it is much more cost effective for musicians to make their own reeds. Musicians use a series of tools to make their reeds including gougers, profilers, and knives. A reed goes through several stages from tube cane to finished form as illustrated in Figure 2. This sequential process is advantageous because the musician can customize the reed to produce the sound they want for a particular occasion. For example, a "thin" reed will produce a higher tone than a "thick" reed. The first major step in making a reed is gouging. Currently there are manual gougers available from Jones Reed and Forrest Music. An example is shown in Figure 3.

Figure 2. Stages of Preparation for Bassoon Reeds

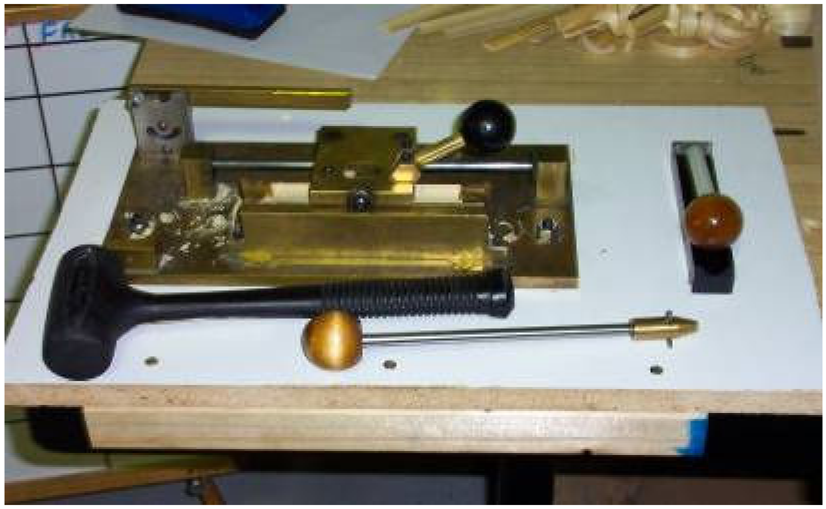

Figure 3. Manual Reed Gouging Apparatus

The first part of problem definition involves a customer interview that entails questions about user needs and experience, device functionality, cost, size, appearance, and expectations for upkeep/replacement parts. Historically, this process results in a problem statement and a set of 'musts' and 'shoulds' that are reported back to the customer within two weeks of the customer interview. The initial problem definition created by this year's design team is given below. Their problem statement captures the essence and opportunity of their problem and includes many well-quantified statements about what the device must accomplish. This result is typical for many of our design teams.

Problem Statement:

Design and build a machine (or set of machines) that can gouge a piece of split cane that can ultimately be shaped into a bassoon reed. The machine(s) must not be labor intensive, time consuming to operate, complicated to learn, or unsafe for laymen. These machine(s) will be located in the UI Music Department and used by a community of faculty and student bassoonists. The device(s) should not compromise the quality of gouged reeds produced by the current method. Ideally, the device(s) could lead to patent opportunities, and possibly the removal of the pre-gouging step.

The Device(s) Must:

Produce gouged reeds 120 mm in length
Produce gouged reeds of 18-20 mm width
Produce gouged reeds of 1.05-1.6mm thickness, in 0.05 mm increments
Produce gouged reeds with an inside radius of 0.5 in
Create a surface finish comparable to 90 grit sandpaper on the inner surface
Accommodate commercially available cane

The Device(s) Should:

Require no more than 10 lb of input force
Weigh less than 20 pounds
Be transportable by one average-sized person
Produce eccentric or concentric reeds
Be configured for operation in less than 10 min
Complete gouging operations in 4-6 min

Axiomatic Design is not intended to replace these initial problem finding activities. Rather, it is intended as a means of highlighting central customer needs and elaborating problem definition by parsing specifications into constraints, functional requirements, and associated design parameters. In the case of the reed gouger, the primary customer need is to efficiently reduce cane to correct dimensions for profiling. Constraints for this problem are essentially the list of 'shoulds' given above along with the requirement that the device be compatible with commercially available cane.

The first 90-minute session with the design team, a number of graduate student mentors and course instructors produced the symmetric tree shown in Figure 4. The tree matrix shown in Figure 5 identifies several areas in the gouging operation where proposed design parameters could compromise meeting functional requirements. In tackling these, it was determined to explore viability of shaping methods first, fixturing second, and safety third.

#	[FR]Functional Requirements	[DP]Design Parameters
0	FR Make reeds	DP Set of mechanisms to make reeds
1	FR Mechanism to cut reeds	DP Guillotine
1.1	FR Position	DP Insert into reed tray and push up against stop with hand
1.2	FR Hold reed	DP Hold down with hand
1.3	FR Cut reed	DP Use handle to chop off reed
2	FR Mechanism(s) to gouge reeds	DP Two mechanisms
2.1	FR pre-gouge	DP "can crusher"
2.1.1	FR position	DP Tray with stops and rails
2.1.2	FR hold	DP Blade has attached roller to hold down reed
2.1.3	FR cut	DP Lever slides blade horizontally over top of reed
2.2	FR gouge	DP ball end mill
2.2.1	FR position	DP reed is inserted into tray with stops at each end
2.2.2	FR set thickness	DP mill bit can be raised or lowered to correct height to yield proper thickness
2.2.3	FR hold	DP rollers on either end of mill bit move horizontally with bit to hold down reed
2.2.4	FR cut	DP 0.5" ball end mill shaves away excess material
3	FR Musician friendliness	DP
3.1	FR Operator feedback: feel of cutting reed	DP
3.2	FR Portability: one person	DP
3.3	FR Training time = 1 hr maximum	DP

Figure 4. Initial Design Decomposition

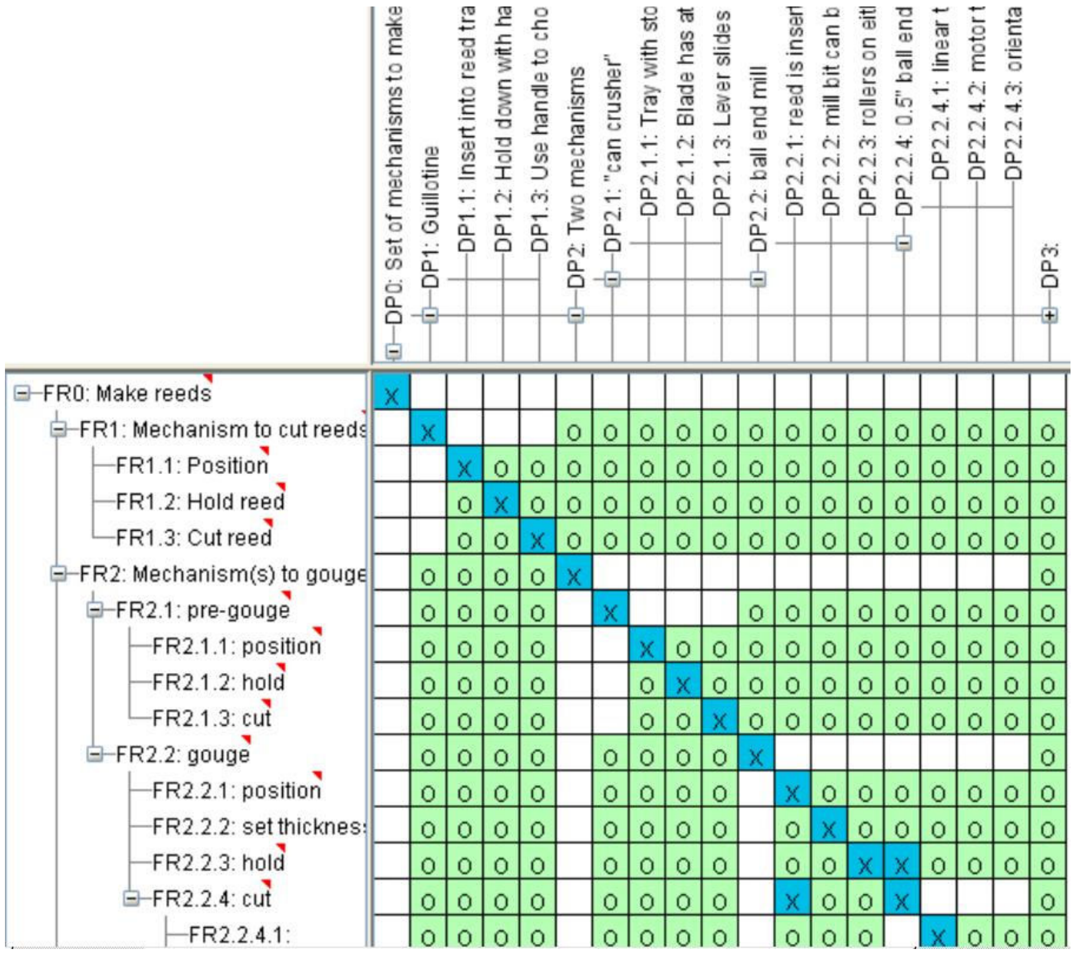

Figure 5. Interaction Between Preliminary DPs and FRs

After three weeks of cutting and sanding research, force analysis, and testing in a wood shop, the group reconvened and produced the result shown in Figure 6. This representation of the design problem is more complete, hierarchical, and decoupled than many past designs. Because of more substantive dialogue with the design teams involved in our pilot, we are also much more aware of the strengths and risks associated with the solution paths these teams are pursuing.

Figures 4-7 are examples of student work and could be improved. We note that in Fig. 4 FR3.1.1 and FR 3.1.2 are not stated in the imperative as suggested in Table 4. However the design intent is clear and the correction is trivial, e.g, "provide easy to use adjustment" and "provide easy to read feedback on current setting". Frequently this kind of correction is not made, unless it is going to have an impact on the design. The decomposition does not have to be optimal for the Axiomatic Design process to be effective in improving the design process and minimizing unnecessary design iterations that can significantly delay the project.

#	[FR]Functional Requirements	[DP]Design Parameters
0	FR Reduce cane to correct dimensions	DP Set of mechanisms to make reeds
1	FR Cut reeds to length	DP Guillotine
1.1	FR Position reed	DP Tray with stop at one end
1.2	FR Hold reed	DP Slot for reed
1.3	FR Cut reed	DP Blade with lever arm and handle
2	FR standard size and shape	DP sanding wheel assembly
2.1	FR prevent motion of reed	DP jigs to hold reed
2.2	FR be easy to operate	DP guide for jig
2.3	FR sand down reed	DP sandpaper
3	FR shape to final dimensions	DP mechanized gouger
3.1	FR set thickness	DP adjustable jig ceiling
3.1.1	FR be easy to adjust thickness	DP screw mechanism
3.1.2	FR be easy to read thickness	DP dial indicator on top of jig
3.2	FR prevent motion of reed	DP fixturing
3.2.1	FR prevent falling	DP leading and trailing blocks
3.2.2	FR prevent tilting	DP leading and trailing blocks
3.2.3	FR prevent lateral movement	DP jig walls
3.3	FR cut	DP router bit assembly
3.3.1	FR cut correct profile	DP 1 1/8" router bit
3.3.2	FR cut smooth surface	DP finish pass
3.3.3	FR protect user from chips and bit	DP cover over bit
3.3.4	FR be easy to manipulate jig	DP guides for jig

Figure 6. Design Decomposition at end of Conceptual Design

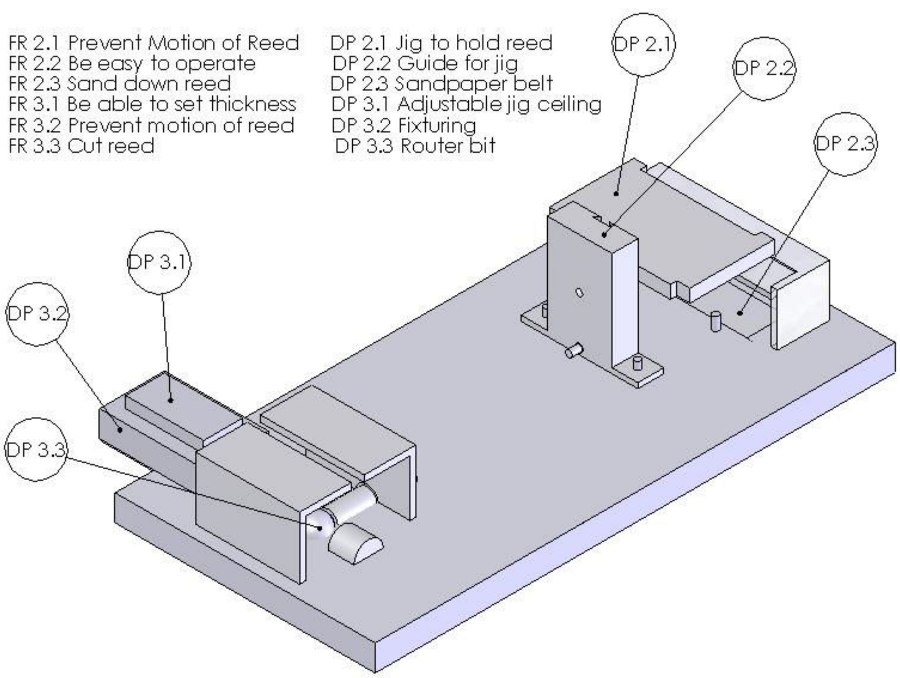

Figure 7. Illustration of DPs after System Integration

5. Conclusions

In this qualitative study, we used a case-study approach to find the issues and benefits associated with application of axiomatic design in a capstone design course. One major finding was that axiomatic design theory correlated with past project experiences. Projects that aligned with the axioms were more successful than projects that did not align with the axioms. A second major finding was that axiomatic design benefited ongoing projects by promoting dialogue between students and instructors during conceptual design, by structuring the process of decomposition, and by reducing the problem of students becoming fixated with a solution concept early in the design process. As we embark on more and more interdisciplinary projects, the Axiomatic Design model presented here and supporting Acclaro software is likely to be helpful in clarifying responsibilities of subteams so that they can proceed toward product realization in a concurrent rather than sequential manner. A third major finding was the value of learning Axiomatic Design in a social context. This is not a trivial practice to master. As a case in point, the research presented in this paper was possible because of a semester-long collaboration between design instructors, an external consultant who provided helpful assistance based on more extensive experience using Axiomatic Design, graduate student mentors, and design team members in a series of workshops, design reviews, and consulting sessions surrounding Axiomatic Design methods.

For the teams currently using Axiomatic Design, we observed a stronger connection between the team's problem definition and conceptual design activities. These teams established functional requirements that were more complete, hierarchical, and independent than other design teams that only generated a set of 'musts' and 'shoulds' for their design. Because of the social interaction promoted by this Axiomatic Design thinking, a broader set of stakeholders was involved in design decision-making and design ideas were more likely to be accepted or rejected on their own merits rather than the force of the personalities presenting different alternatives.

The issues about using Axiomatic Design in a capstone course relate to how it should be implemented, not whether it is valuable in measuring design quality. This was addressed by bringing someone in with experience in the use of axiomatic design. In this paper we have attempted to illuminate and address these issues so that others can implement Axiomatic Design more successfully without relying on an experienced consultant.

References

1. Suh, N. 1990. The Principles of Design, Oxford University Press, New York.

2. Suh, N. 2001. Axiomatic Design: Principles and Applications. Oxford University Press, New York.

3. Tate, D. and Nordlund, M. 1996. Design Process Roadmap as a General Tool for Structuring and Supporting Design Activities, Proceedings of the Second World Conference on Integrated Design and Process Technology (IDPT-Vol. 3), Society for Design and Process Science, pp 97-104.

4. Todd, R., Magleby, S., Sorensen, C., Swan, B., and Anthony, D. 1995. A Survey of Capstone Engineering Courses in North America. *Engineering Education*: 165-174.

5. McKenzie, L., Trevisan, M., Davis, D., and Beyerlein, S. 2004. Capstone Design Courses and Assessment: A National Study. Proceedings of the Annual Meeting of the American Society for Engineering Education, Salt Lake City, June 2004.

6. Ulmann, D. 1992. The Mechanical Design Process. McGraw-Hill, New York.

7. Pahl, G. and Beitz, W. 1996. Engineering Design, 2nd Edition. Springer-Verlag, New York.

8. Dym, C. and Little, P. 2000. Engineering Design: A Project-Based Introduction. John Wiley & Sons, New York.

9. Dominick, P., Demel, J., Lawbaugh, W., Freuler, R., Kinzel, G., and Fromm, E. 2001. Tools and Tactics of Design. John Wiley & Sons, New York.

10. Ulrich, K. and Eppinger, S. 2004. Product Design and Development, 3rd Edition. McGraw-Hill, New York.

11. Cordon, D., Clarke, E., Westra, L., Allen, N., Cunnington, M., Drew, B., Gerbus, D., Klein, M., Walker, M., Odom, E., Rink, K., and Beyerlein, S. 2002. Shop Orientation to Enhance Design for Manufacturing in Capstone Projects, Proceedings of the Frontiers in Education Conference, October 2002.

12. Gerbus, D., Odom, E., and Beyerlein, S. 2001. Applying Theory of Constraints to Solicit Feedback and Structure Improvements to a Capstone Design Experience, Proceedings of the American Society for Engineering Education Annual Conference and Exposition, June 2001.

13. Drew, R., DuBuisson, A., Milligan, B., Williams, J., Beyerlein, S., Odom, E., and Rink, K. 2003. Early Development of Capstone Design Teams through Graduate Student Mentoring and Team Building Activities, Proceedings of the American Society for Engineering Education Annual Conference and Exposition, June 2003.

Author Biographies

EDWIN ODOM has taken an active interest in the ME Machine Shop as a key element in design education since joining the University of Idaho fourteen years ago. Dr. Odom maintains an avid interest in the literature of creativity and management and is especially well versed on the subjects of engineering mechanics and machine design. He was recognized for his role in development of the Idaho Engineering Works by a university teaching award in 1998.

STEVEN BEYERLEIN is professor of Mechanical Engineering at the University of Idaho, where he coordinates the capstone design program and regularly participates in ongoing program assessment activities. For these efforts he won the UI Outstanding Teaching Award in 2001. He received a Ph.D. in M.E. from Washington State University in 1987. His research interests include catalytic combustion systems, application of educational research methods in engineering classrooms, and facilitation of faculty development activities.

CHRISTOPHER BROWN is a professor at Worcester Polytechnic Institute. He received his Ph.D. in 1983 from the University of Vermont. He teaches courses on surface metrology, design and analysis of manufacturing processes, axiomatic design, and the technology of alpine skiing. He is the founder and director of WPI's Haas Technical Education Center for CNC machining and WPI's Surface Metrology Lab. He is an International Director of the Society for Manufacturing Engineers (SME), a corresponding member of CIRP (International Institution for Production Engineering Research), chair of the subcommittee on fractal methods for ASME/ANSI B46 Committee on Surface Texture, and chair of the SME MTA (Machining Technology Association) advisory board. He has applied axiomatic design to a wide range of problems including crew survivability systems for an orbital space plane, scanning laser microscopes, and ski bindings.

DANIEL DREW is a senior in Mechanical Engineering at the University of Idaho, with an expected graduation date of May 2005. He is currently in the Capstone Design course working on the reed gouger project discussed in this paper. He plans on continuing his education at a graduate school to be determined.

LLOYD GALLUP is a senior in Mechanical Engineering at the University of Idaho, with an expected graduation date of May 2005. He is currently in the Capstone Design course working on the reed gouger project discussed in this paper. He plans to pursue a graduate degree at the University of Idaho in the field of quality engineering.

SAM ZIMMERMAN is a graduate student in Mechanical Engineering at the University of Idaho, where has mentored numerous capstone design teams and has taught Principles of Lean Manufacturing. He received his B.S.M.E. from the University of Idaho in 2003. His research interests include acoustic optimization of intake manifolds.

JEREMY OLBERDING received his B.S.M.E. from the University of Idaho in 2003. He is currently pursuing graduate work focusing on catalytic combustion systems. He is an active member of the Idaho Engineering Works that provides mentoring for capstone design students and stewardship of departmental design resources.