DATABASE SUPPORT FOR REVERSE ENGINEERING, PRODUCT TEARDOWN, AND REDESIGN AS INTEGRATED INTO A MECHANICAL ENGINEERING COURSE

Mark Snider, Joshua D. Summers, Gregory M. Mocko, and Sudhakar Teegavarapu
Department of Mechanical Engineering
Clemson University
Clemson, SC 29634-0921

Abstract

This paper introduces a database to aid with the specific tasks of product teardown and redesign. The database contains entry fields for relevant data, such as physical parameters, function details, connectivity, and failure modes and effects analysis. The initial creation of the database was to assist with a reverse engineering and product benchmarking project for automotive applications. This particular project's motivation was to reduce the mass of the headlight module; the physical parameters were recorded with this in mind. The function data fields were structured so that the user was forced to follow a predefined terminology, thus diminishing inconsistency between users. Failure Modes Effects Analysis (FMEA) was implemented into the teardown procedure to elicit novel concepts and to force the user to pay attention to detail during the teardown. Evolution of the relational table structure of the database made it more robust, thus providing more valuable data comparisons. The database has also been implemented in a university class to provide a teaching mechanism for reverse engineering and redesign.

Keywords: *database, reverse engineering, benchmarking, product teardown*

Introduction

The popularity of reverse engineering methods is growing due to the changes of the global economy. Methods for reverse engineering are starting to emerge to satisfy this demand. However, there are still gaps within these toolsets that remain unfilled. It is apparent that the practice of reverse engineering in some form has been conducted in industry for a long time. This can be attributed to three areas: technological, economic, and political[1]. Two motivations identified for reverse engineering is to benchmark competitor products to ensure that best practices are maintained and to recreate the design rationale used to design legacy products where there is a lack of documentation. However, most documented reverse engineering methods have been targeted at software problems, which typically deal with the task of understanding legacy programs (e.g.[2,3]), while little attention has been given to hardware applications. With the global market place of today, companies are turning more and more to reverse engineering practices [1]. As a result, some reverse engineering methods are starting to emerge[4], providing a general overview of the process. Specifically, reverse engineering and product teardown are increasingly being used as a teaching tool for mechanical engineering [5-8].

This paper describes a reverse engineering database that was originally developed to support two industry sponsored research projects and has been restructured to support reverse engineering in an undergraduate mechanical engineering course. The paper begins with a discussion on the database and concludes with the lessons learned with respect to integration as a support tool for coursework.

Motivation

While conducting a reverse engineering and benchmarking project of automotive parts it became clear that the teardown process needed streamlined. The initial procedure consisted of a series of teardown documents that were filled out for each individual assembly/component [9]. Each form contained data pertaining to each individual assembly and component. However, when the teardown was completed, post-processing the data into comparative reports was difficult with the current hard copy approach. This required manual extraction of the data from each teardown sheet. Once the pertinent data was extracted it was reentered (or copied) into spreadsheets to calculate mass per function and other comparative queries. It was determined that something needed to be developed to assist with the teardown procedure.

In this paper, a database tool is offered to aid with product teardown and experimentation as a step in the reverse engineering and redesign method [4]. Outlined are the details of the database as it was created with the specific application of the reverse engineering and benchmarking project for automotive applications (headlight modules and seats), and the ensuing evolution of the database. Justification is given for the reasoning behind the structure of the database teardown forms, as well as the data fields contained within the forms.

Data Entry

It was decided that an electronic database be created in order to efficiently document and subsequently manage the desired data for a reverse engineering product teardown. The implementation of an electronic aid is targeted to allow the teardown process to be carried out in a more effective an efficient manner [10]. When creating the database, it was elected to provide a teardown sheet for every assembly/component contained within the system, this done to aid with understanding the product, and as a possible guide for reassembly of the product[9]. Each teardown sheet contained data fields for all relevant information. The first generation database teardown form design was based on the data template for disassembly and experimentation [9]. As the project progressed, the teardown forms continued to expand in order to accommodate additional information that was not originally foreseen to be of value, such as: part type (assembly/component), and failure modes and effects analysis. The first generation database version contained data fields that were either text fields or number fields. Specifying the data types in the fields helped restrict data from being entered incorrectly [11]. The last version of the first generation database teardown sheet can be seen in Figure 1, this version has expanded dramatically over the data template for disassembly and experimentation [9]. Most of the data fields have expanded, but still continue to be formatted as text and number fields; there were some implementation of some pull-down menus where applicable.

Table 1 contains specification for all data fields contained with the teardown form. However, some fields show need for modification and or implementation of pull down menus to help eliminate inconsistency between engineers conducting the teardowns in the future. For example, function details are prone to inconsistency between engineers due to varying vocabulary.

A problem that arose when entering data into the teardown sheets was inconsistency in some of the text fields. Since there are typically multiple individuals conducting the teardowns, it should be expected that there could be some inconsistency in the manner each individual identified certain details. The most troublesome fields were those of the function classes; each teardown sheet contains three fields that identify the primary, secondary, and tertiary functions of each subassembly and component. Consistently identifying functions can be difficult especially when dealing with multiple persons. Therefore, when beginning the project it was decided that a predefined function vocabulary would be used, this is detailed in Table 2 [12]. This still allowed some inconsistency between individuals. In order to compensate for this inconsistency the function fields were separated into two parts; the first consisting of function class, and the second function details.

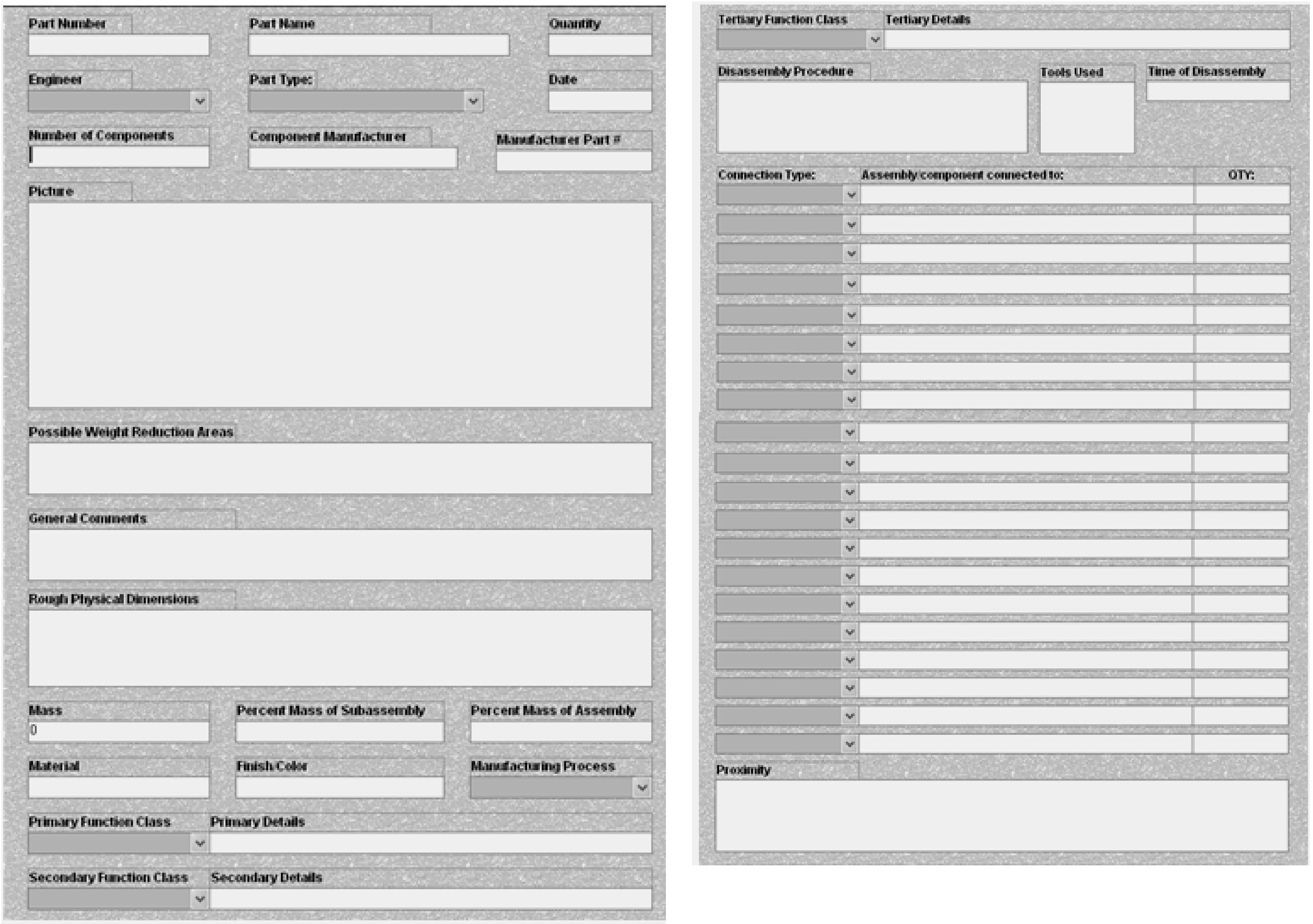

Figure 1: 1st Generation Teardown Forms.

The function class fields are structured so as to utilize a pull down menu; thus compelling the users conducting the teardown to identify one of the eight defined function classes: branch, channel, connect, control magnitude, convert, provision, signal, and support[12]. Figure 2 shows an example of the function section of the teardown form. This particular form is that of a headlight module high-beam reflector. This example illustrates that the tertiary function of the high-beam reflector is to increase light intensity of light emitted from the high-beam light module, which is categorized under the control magnitude function class. Since there are many underlying functions of each headlight module it is difficult to group the components based on functionality. Therefore, by identifying the function class allows each component to be grouped in a more manageable fashion. Not only does the addition of the pull down menu create more consistent function records, it also streamlines the data entry, thus saving time; based on anecdotal experience with the database on nine products in two domains (6 headlights, 2 seats).

Table 1: Data Field Details.

Field Name:	Data Type:	Description:
Part Number	Text	number based upon numbering method
Part Name	Text	short descriptive part name
Quantity	Number	number of duplicate assemblies or components
Engineer	Pull down	user whom conducted documentation of given assembly/ component
Part Type	Pull Down	designates whether the part is an assembly, component, or hardware
Date	Date/Time	day teardown occurred
# of Components	Number	number of components that make up a given assembly
Component Manufacturer	Text	original component manufacturer if applicable
Manufacturer Part #	Text	original component part number
Picture	OLE object	pictures taken of assembly/component during teardown
Possible Mass Reduction Areas	Text	initial hypothesis of potential mass reduction ideas
General Comments	Text	interesting comments that could be of potential use
Rough Physical Dimensions	Text	any relevant physical dimensions
Mass	Number	mass of assembly/component in grams (g)
% Mass of Subassembly	Number	percent mass of the assembly/component with respect to the assembly it is part of, if applicable
% Mass of Assembly	Number	percent mass of the assembly/component with respect to the complete assembly
Material	Text	material the component is
Finish / Color	Text	finish or color of component
Manufacturing Process	Text	manufacturing process used to produce the assembly/component
Function Class[3]	Pull Down	function definition class
Function Details[3]	Text	details specifying the function using vocabulary from [12]
Disassembly Procedure	Text	brief description of steps required to remove assembly/component form overall assembly
Tools Used	Text	listing of all tools required to remove assembly/component
Time of Disassembly	Number	amount of time required to remove given assembly/component
Connection Type[20]	Pull Down	type of connection given assembly/component has to another assembly/component
Connected to Assembly/Component[20]	Text	part number and part name of assembly/component in which the given assembly/component is connect to
QTY. [20]	Number	number of a certain connection type the given assembly/component has
Proximity	Text	brief description of any potentially hazardous neighboring assemblies/components
Failure Mode[3]	Text	potential type of failure
Effects of Failure[3]	Text	potential effects of a given failure
Current Control[3]	Text	design variables
Severity[3]	Number	numeric rank of the severity of effects from potential failure
Likelihood of Failure[3]	Number	numeric rank of the chance of potential failure mode of happening
Likelihood of Detection[3]	Number	numeric rank of the chance of detecting potential failure mode prior to failure

* Note: Superscript denotes the number of times a field is repeated in the teardown form

Class (Primary)	*Secondary*	*Tertiary*	*Correspondents*
Branch	Separate		Isolate, sever, disjoin
		Divide	Detach, *isolate*, release, sort, split, disconnect, subtract
		Extract	Refine, filter, purify, percolate, strain, *clear*
		Remove	Cut, drill, lathe, polish, sand
	Distribute		Diffuse, dispel, disperse, dissipate, diverge, scatter
Channel	Import		Form entrance, *allow*, input, *capture*
	Export		Dispose, eject, *emit*, empty, *remove*, destroy, eliminate
	Transfer		Carry, deliver
		Transport	Advance, lift, move
		Transmit	Conduct, convey
	Guide		Direct, shift, steer, straighten, switch
		Translate	Move, relocate
		Rotate	Spin, turn
		Allow DOF	*Constrain*, unfasten, unlock
Connect	Couple		Associate, connect
		Join	Assemble, fasten
		Link	Attach
	Mix		Add, blend, coalesce, combine, pack
Control Magnitude	Actuate		Enable, initiate, start, turn-on
	Regulate		Control, equalize, limit, maintain
		Increase	*Allow*, open
		Decrease	Close, delay, interrupt
	Change		Adjust, modulate, *clear*, demodulate, invert, normalize, rectify, reset, scale, vary, modify
		Increment	Amplify, enhance, magnify, multiply
		Decrement	Attenuate, dampen, reduce
		Shape	Compact, compress, crush, pierce, deform, form
		Condition	Prepare, adapt, treat
	Stop		End, halt, pause, interrupt, restrain
		Prevent	Disable, turn-off
		Inhibit	Shield, insulate, protect, resist
Convert	Convert		Condense, create, decode, differentiate, digitize, encode, evaporate, generate, integrate, liquefy, *process*, solidify, transform
Provision	Store		Accumulate
		Contain	*Capture*, enclose
		Collect	Absorb, consume, fill, reserve
	Supply		Provide, replenish, retrieve
Signal	Sense		Feel, determine
		Detect	Discern, perceive, recognize
		Measure	Identify, *locate*
	Indicate		Announce, show, denote, record, register
		Track	Mark, time
		Display	*Emit*, expose, select
	Process		Compare, calculate, check
Support	Stabilize		Steady
	Secure		*Constrain*, hold, place, fix
	Position		Align, *locate*, orient
Overall increasing degree of specification →			

Table 2: Function Vocabulary [12].

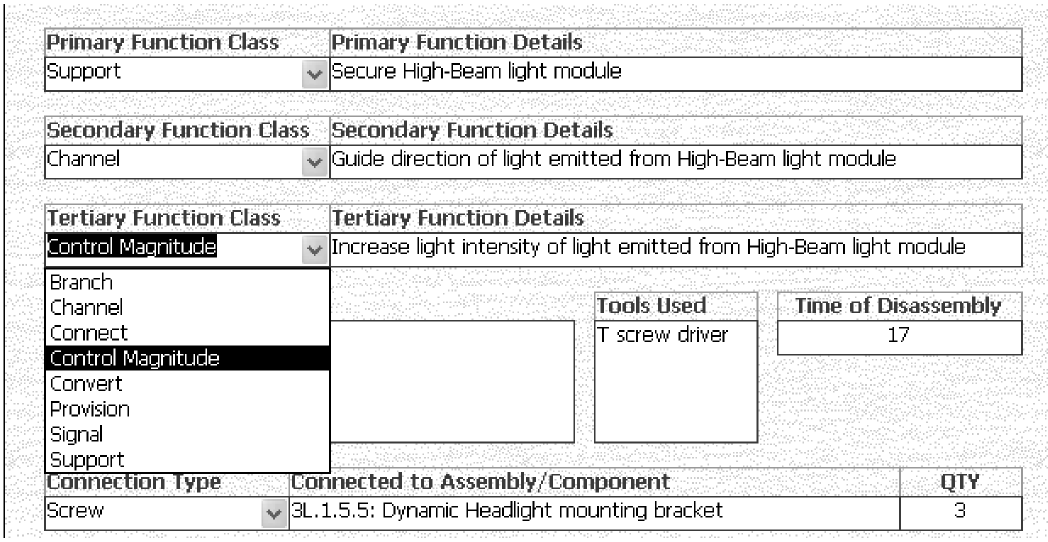

Figure 2: Function Example for Headlight Module High-Beam Reflector.

Other fields that were modified into pull down menus include: engineer, part type, and connection type. The engineer field is a text field that documents which individual conducted the teardown for that particular assembly or component; this also imposes accountability on the engineer for that particular teardown. Forcing ownership, helps insures that care is given and quality work is being done. Formatting the field as a pull down menu helps save the engineer valuable time throughout the teardown, by not having to redundantly type his or her name several number of times. Another field that incorporates the use of a pull down menu is that of the connection type. Twelve different options for connection type are identified, these include: ball joint, electrical, fuse, glue, hinge, interference, press, screw, slide, snap, twist, and weld. There is also twenty entries for connections on each teardown form, the reason for such large number of these fields is to accommodate certain assemblies or components that may be connected to a variety of different components, and have multiple connections to a single component. An example of which is seen in Figure 3. The pull down menu provides a reliable method of cataloguing all possible connections a component might have. Special attention is given to the connections because of the importance of generating connectivity charts, and parallel function models.

An additional data field that incorporates the pull down menu is that of part type. The part type field was one of the last additions made to the first generation database teardown forms, and was created for post processing purposes. The part type field has two pull down options: assembly, and component. Since a teardown form is created for each assembly, as well as each component contained within each assembly; it is necessary to distinguish between the two possibilities. For instance if the sum of masses for a certain function class were to be calculated, a query could be made that only sums components that fit into that function class so as not to redundantly calculate a component contained within an assembly. The pull down menu eliminates any typing error that may result, and thus eliminates any possibility that a components mass is not included.

The last version of the first generation database teardown form includes a section for failure mode and effect analysis (FMEA)[13]. Each form contains data fields for three failure modes, with consequent fields detailing the effects of failure, and current control, as seen in Figure 4. Each failure mode is then analyzed and given a ranking for severity, likelihood of failure, and likelihood of detection. Conducting FMEA on each assembly and component during the teardown procedure, introduces a new method of performing FMEA, which is historically done after teardown was completed.

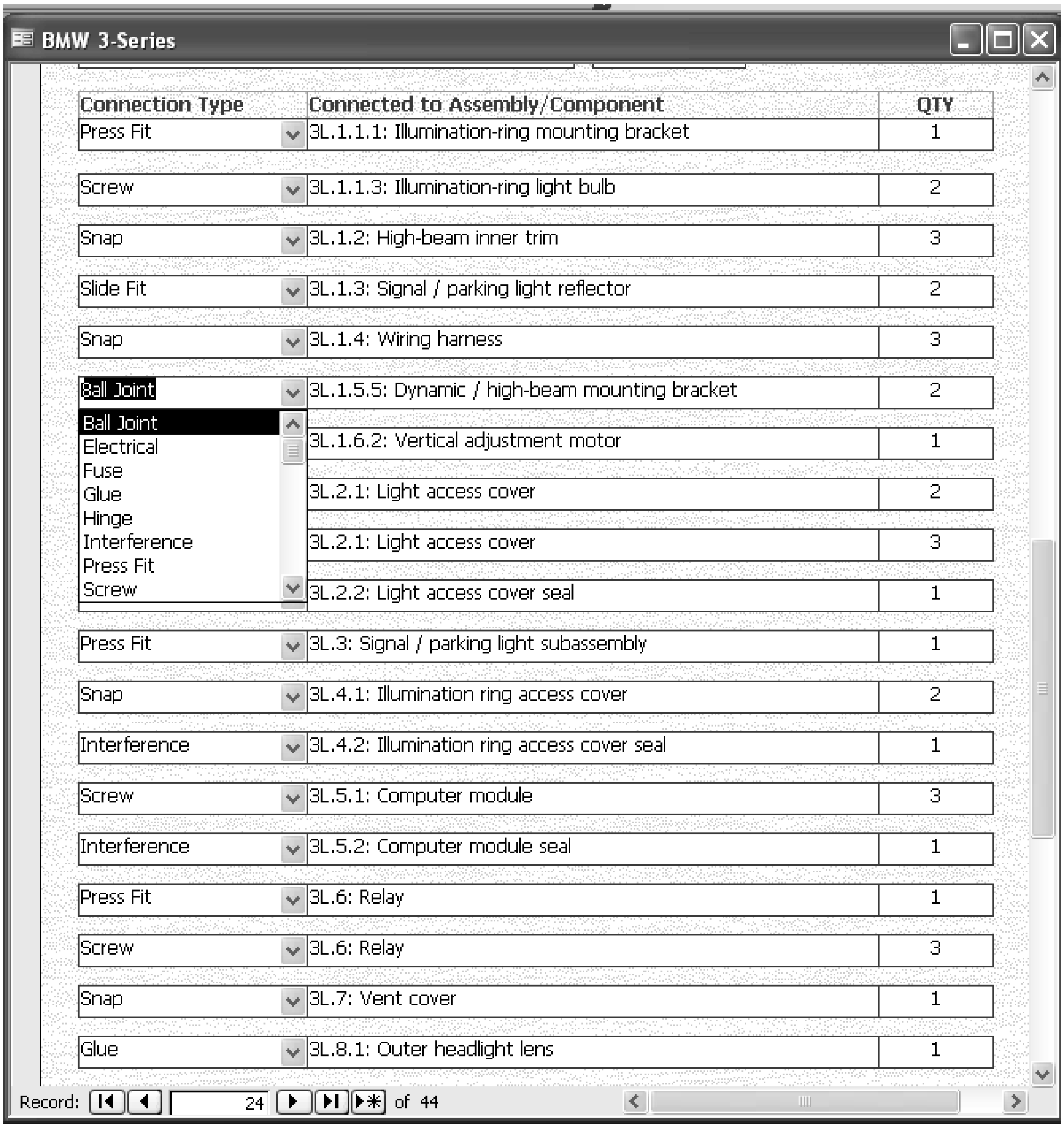

Figure 3: Teardown Form Connection Section.

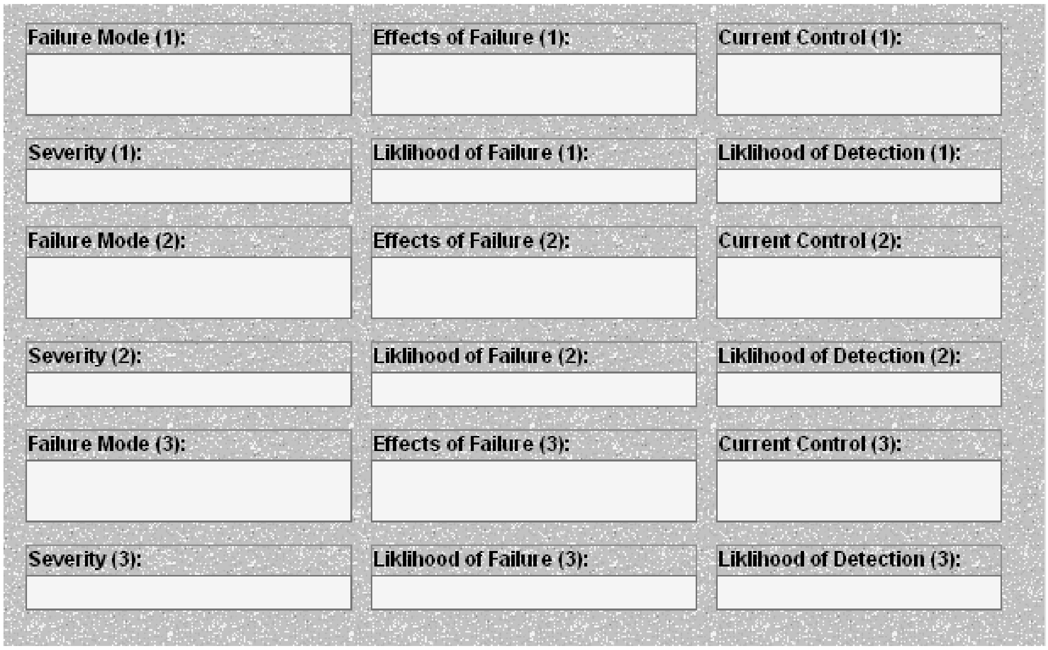

Figure 4: Failure Modes and Effects Analysis Section.

Performing FMEA during the teardown allows for a more clear analysis, given that all relative data of the component being scrutinized is fresh in the mind of the engineer rather than rehashed information. Further, experience with using the FMEA section of the database suggest that it helps focus the engineer's attention to the details, thus helping to develop a better technical picture of the assembly/component being analyzed.

Post Processing

Summaries:

The motivation for creating the database was to reduce the time required for teardown, and post teardown data analysis. Prior to creation of the database after a teardown was completed all summary documents were created manually by extracting data from each individual teardown sheet. Even though this was a rather remedial task to perform, it inevitably took valuable time that could be better spent on further evaluation.

When initially creating the database, tables were created for each individual headlight module that contains data fields for all information collected from the teardown procedure. Teardown forms were then created to serve as a means of populating the tables, as well as summary documents for each assembly/component. An advantage of the database is the ability to extract a variety of information at any given time. For instance a bill of materials (BOM) is automatically populated as the teardown proceeds. An example of an extraction of a BOM can be seen in Figure 5. This type of procedure is also of benefit when acquiring summary statistics regarding entities such as: material, function, and manufacturing processes. The time required to generate each of these summary tables is roughly the same amount of time it would takes to enter information of one assembly or component manually. An added benefit of the database is that when each of these tables is generated the software package provides the option of sorting descending or ascending. Rather than sorting through each teardown form manually, which is quite cumbersome; the database allows for automatic identification.

Queries:

While the database provides a convenience when it comes to generating summary tables, the real power lies within the query function. Queries allow the user to retrieve, change, add

BOM

Part Number	Part Name	Quantity	Mass
6L.0	Headlamp Module Assembly	1	3084
6L.1.1	Dynamic/Secondary Light Assembly	1	978.25
6L.1.1.1	Secondary Light Bulb Assembly	1	14.95
6L.1.1.2	Secondary Light Bracket	1	48.5
6L.1.1.2-H	T15 Torx Screw	3	6.25
6L.1.1.3	Dynamic Light Reflector Ring	1	22
6L.1.1.4	Xenon Light Bulb Assembly	1	114.3
6L.1.1.5	Dynamic Light Module	1	772.25
6L.1.2	Vertical Adjustment Assembly	1	185.95
6L.1.2.1	External Vertical Adjustor	1	3.4
6L.1.2.2	Fiber-Optic Guide	1	6.5
6L.1.2.3	Vertical Adjustment Motor	1	90

Figure 5: Generic BOM Extraction from a Headlight Module Teardown.

to, and analyze data from one or more tables or queries, in response to user defined criteria. One particular method used is to identify redundant functions, for instance a query is created that searches through all the teardown forms for duplicate function details. If any are identified it may be possible to eliminate components completely in the redesign of the system. Another advantage the queries provide is the ability to sort and calculate desired values. Figure 6 shows a table of the summation ofmasses of components with respect to each components primary function. The query identifies only components not assemblies, from the teardown forms, grouping them according to primary function and then summing the masses.

Function : Select Query

Primary Function Class	SumOfMass
Channel	384.39
Control Magnitude	1058.45
Convert	791.93
Signal	21.66
Support	1175.44

Record: 4 of

Figure 6: Function Class Mass Summary.

Discussion

The initial project from which the motivation for the database tooling support for product teardowns stemmed was that of a headlight module reverse engineering and redesign project. The project consisted of reverse engineering six headlight modules and five car seat modules as found in multiple vehicles from multiple manufacturers, the emphasis being on reducing the overall mass. It was quickly found that a significant amount of time could be saved if the redundancy of data reentry could be eliminated, as well as create a way to automatically perform the necessary calculations. Calculations of interest include: summation of weights with regard to function, material, and manufacturing process. Queries were particularly versatile when identifying possible mass reduction on a function level. This allowed for quick recognition of which function contributes the most to the overall mass of the headlight modules. The same method is used with respect to material instead of function. Determining the amount of mass due a given material could and did lead to a material change to reduce the overall mass of the headlight module. Also of interest was the identification of the largest component weight contributors. Given that the primary objective of the project is to reduce mass of the headlight or car seat modules it is favorable to identify the primary mass contributors to the overall assembly.

Second Generation

Evolution of the database has occurred since the initial lightweight headlight module reverse engineering project was concluded. The second generation of the database contains similar data fields, but now uses a more robust underlying relational schema. An entity relationship (ER) diagram of the second generation database can be seen in Figure 7. The relational structure of the second generation database allows for cross model comparison. This allows users to compare the aforementioned summary tables such as primary function class, material, manufacturing process, etc. across models. This of course can be done manually, but requires a substantial amount of time. The automatic comparison is useful for benchmarking purposes, to elicit best practices. This was one of the primary weaknesses of the first generation database.

It has been documented that increasing hands-on engineering/redesign in design classes leads to substantial improvements in course ratings[8]. Therefore, the second generation database is currently being utilized in a project for a Design for CAM/Design for Manufacture course at Clemson University. The project consists of reverse engineering a small consumer product (electric drill, blender, mixer, or knife), with redesign emphasis placed on manufacturing/functional considerations. The second generation is structured in such a way that teardown data can be entered for all of the four consumer products into the same database. This allows for cross comparison over the entire range of products, or comparison for strictly one product.

Student experience with using this database in two sections of the course (Fall 2005 and Spring 2007) suggests that the emphasis on complete documentation of the product tear down was critical to the successful understanding of how the different consumer products function.

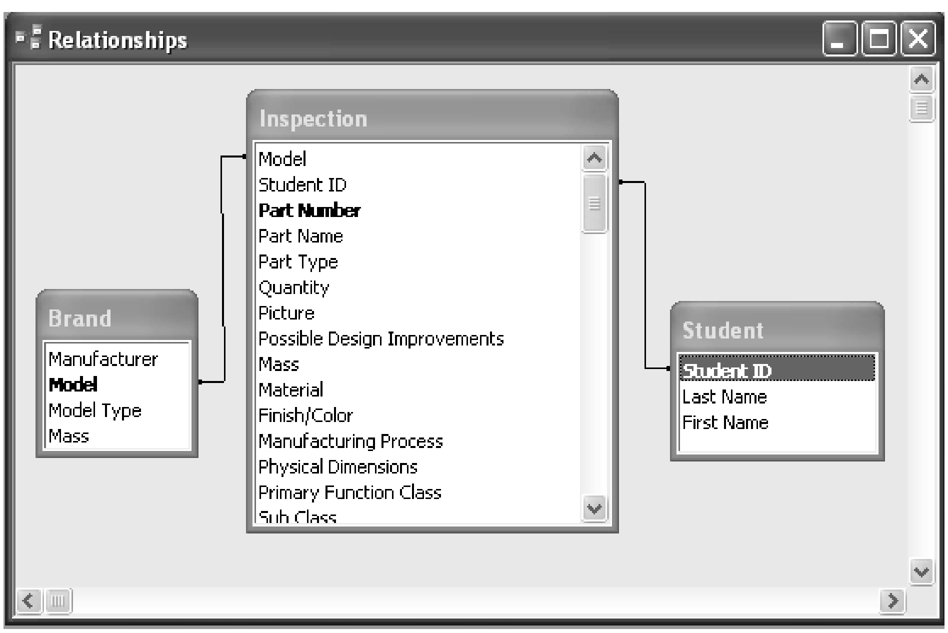

Figure 7: Second Generation ER Diagram.

Specific comments from the students include:

- "While the database was tedious, it was useful to force me to look at all interactions in the knife. We were able to tie functions to components to understand different possible failures." (Fall 2005)

- "I would suggest teaching this database sooner in the class – use it for the design for assembly project." (Spring 2007)

- "We used the FMEA section to see where we can improve the design. I mentioned this on an interview and they loved it!" (Spring 2007)

While these comments, and others, are anecdotal, it appears that the students recognize this tool as useful in the coursework. Further integration of this database (and subsequent evolutions) are being considered for additional courses in the undergraduate curriculum.

Future Work

Even though the creation of the database has aided in the analysis and identification of mass reduction potential, many possibilities still exist. Failure mode and effect analysis (FMEA) is a tool used to predict every possible failure mode of a product, and effects that failure mode presents. An FMEA query is to be created that calculates risk for each teardown form and sorts the results in an ascending order. Historically FMEA has been conducted to determine problematic areas that may need redesigned to resist failure. However, a unique approach is being used for this particular circumstance; instead of looking for problematic areas, overly safe areas are targeted. This allows recognition of areas where possible mass reduction could occur without compromising the dependability of the headlight module. A reason the connection portion of the teardown forms provide such a detailed level is to aid in the creation of a connectivity model of each headlight module. At the present moment this is done manually, an example of which can be seen in Figure 8, but is a quite burdensome task, since there can be numerous components in a complete assembly. It is a goal to automatically generate this connectivity model from the teardown forms. If it can be done it would aid in the systematic analysis of the product module. For example if a certain component were to be modified in order to reduce the mass, one could easily see what other components would also have to be modified to accommodate the original change. In accordance with the connectivity graph and the component function fields, current work is being done to generate a function structure[14].

This paper demonstrates that design tools and computational aids developed in the course of engineering research efforts can be successfully integrated in the undergraduate curriculum. Important implications from this are that:

1. students are exposed to cutting edge engineering tools and applications,

2. faculty can use the introduction of these tools in the classroom as a validation testbed,

3. students can provide invaluable feedback to the tool developers with respect to usability, and

4. student learning can be enhanced through the same type of project work that they will experience in industry.

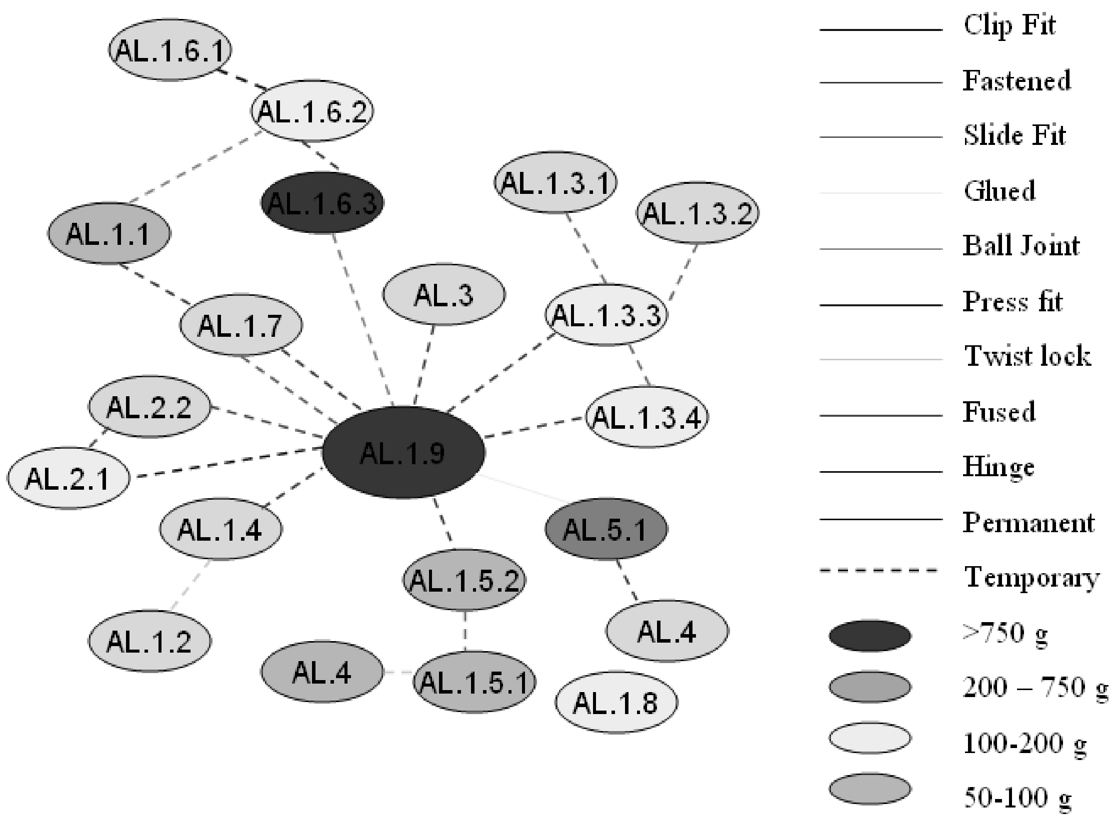

Figure 8: Connectivity Graph for a Headlight Module.

References

1. K. Ingle, *Reverse Engineering*. New York, NY: McGraw-Hill, 1994.

2. R. Chiang, T. Barron, and V. Storey, "Reverse Engineering of Relational Databases: Extraction of an EER model from a Relational Database," *Data & Knowledge Engineering*, vol. 12, no. 2, pp. 107-42, 1994.

3. J. Perez, I. Ramos, V. Anaya, J. Cubel, F. Dominguez, A. Boronat, and J. Carsi, "Data Reverse Engineering of Legacy Databases to Object Oriented Conceptual Schemas," *Electronic Notes in Theoretical Computer Science*, vol. 74, no. 4, pp. 13 pages, 2002.

4. K. Otto and K. Wood, "Product Evolution: A Reverse Engineering and Redesign Methodology," *Research in Engineering Design*, vol. 10, no. 4, pp. 226-43, 1998.

5. S. Sheppard, "Dissection as a Learning Tool," proceedings of the *FIE Conference*, Nashville, TN, 1992.

6. R. E. Barr, P. S. Schmidt, T. J. Krueger, and C. Y. Twu, "An Introduction to Engineering Through an Integrated Reverse Engineering and Design Graphics Project," *Journal of Engineering Education*, vol. 89, no. 4, 2000.

7. D. Jensen, M. Murphy, and K. Wood, "Evaluation and Refinement of a Structured Introduction to Engineering Design Course Using Student Surveys and MBTI Data,"

proceedings of the *ASEE Annual Conference and Exposition: Engineering Education Contributing to US Competitiveness*, Seattle, WA, 1998.

8. D. Jensen and K. Wood, "Incorporating Learning Styles to Enhance Mechanical Engineering Curricula by Restructuring Courses, Increasing Hands-On Activities, and Improving Team Dynamics," proceedings of the *2000 ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, 2000.

9. K. Otto and K. Wood, *Product Design - Techniques in Reverse Engineering and New Product Design*, 1 ed. Upper Saddle River, NJ: Prentice Hall, 2001.

10. S. Pugh, *Total Design*. Wokingham, UK: Addison-Wesley, 1991.

11. J. Cronan, V. Anderson, and B. Anderson, *Microsoft Office Access 2003 Quicksteps*. Emeryville, CA: McGraw-Hill/Osborne, 2004.

12. J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," *Research in Engineering Design*, vol. 13, no. 2, pp. 65-82, 2001.

13. J. Wang, *Engineering Robust Design with Six Sigma*. Upper Saddle River, NJ: Prentice Hall, 2005.

14. G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*. New York, NY: Springer-Verlag, 1996.

Biographical Information

Mark Snider is an product development engineer with ForceProtection. He earned his MS in Mechanical Engineering from Clemson University in 2006, working in the Automation in Design Group under Dr. Joshua D. Summers. He earned his BS in 2004 from Clemson University. Mr. Snider’s graduate research was sponsored by BMW.

Joshua D. Summers, Associate Professor in Mechanical Engineering at Clemson University, leads the Automation in Design Group. Dr. Summers earned his Ph.D. in Mechanical Engineering from Arizona State University in 2004 researching design automation. Dr. Summers received his BSME and MSME from the University of Missouri-Columbia, working on VR-based submarine design. Dr. Summers has worked at the Naval Research Laboratory (VR Lab and Naval Center for Applied Research in Artificial Intelligence) and served on the Foreign Relations/Armed Services staff of Senator John D. Ashcroft. Dr. Summers’ research has been funded by NASA, NSF, US Army TACOM, BMW, Michelin, General Motors, Wright Metal Products, Hartness International, and others. Dr. Summers’ areas of interest include collaborative design, knowledge management, and design automation with the overall objective of improving design through collaboration and computation.

Gregory M. Mocko joined the Department of Mechanical Engineering at Clemson University in 2006 after the completion of his Ph.D. at the Georgia Institute of Technology. He received a B.S. degree in Mechanical Engineering and Materials Science from the University of Connecticut in 1999 and a M.S. degree in Mechanical Engineering from Oregon State University in 2001. His teaching and research interests lie in the general areas of engineering design with an emphasis on engineering information management in engineering design. Dr. Mocko's research group focuses on the development of advanced representations, methods, and tools for facilitating the integration and exchange of information. Dr Mocko’s research has been funded by Hartness International, NSF, BMW, and US Army TACOM.

Sudhakar Teegavarapu is a doctoral student in the Automation in Design Group at Clemson University, studying the development of engineering design methods under the direction of Dr. Joshua D. Summers. His graduate research has been funded by BMW on three distinct development projects.