

System-Level Design: A Missing Link?*

DURWARD K. SOBEK, II

*Department of Mechanical and Industrial Engineering, Montana State University,
220 Roberts Hall, Bozeman, Montana, 59717-3800, USA. E-mail: dsobek@ie.montana.edu.*

System-level design is proposed as the set of activities that aid design engineers in transitioning from highly abstract conceptual ideas to detailed designs. It is characterized by exploration of and decisions about the arrangement of components and subsystems, and design of interfaces. While these activities are sometimes included in engineering design frameworks, it has not been well studied. And yet, the results from three analyses of empirical data derived from mechanical engineering student design projects suggests that such activities seem to have a significant impact on independently assessed outcome measures, even though they comprise a small percentage of student effort. This further suggests that system-level design may be a high leverage activity worthy of further investigation. This paper defines system-level design, summarizes the research methods used and analyses of the empirical data, and presents a tool to aid in the system-level design phase.

Keywords: design subsystems; design frameworks; design decisions.

INTRODUCTION

TOYOTA MOTOR CORPORATION is arguably the most successful automotive company in the world, outperforming its US competitors on many fronts, including product quality and engineering productivity [1]. Prior research on Toyota's product development practice has revealed a number of striking differences from those of US automakers; however, from a design process standpoint, the differences are relatively few [2]. All carmakers study the market and develop many styling alternatives. All gradually eliminate alternatives until a final design remains. All then engineer the design, conduct manufacturability reviews and crash simulations, release engineering drawings for prototype, and create and test prototype vehicles. They all use computer-aided design and analysis tools with similar capabilities.

Interestingly, though, Toyota has a phase in body engineering that others do not: 'kozokeikaku,' roughly translated as 'design structures plan.' From the concept design, rather than proceeding immediately to detailed engineering drawings and analysis, body engineers develop the key cross-sections of the sheet metal parts, and specify key interfaces. They create a document (the 'kozokeikaku') incorporating these preliminary design plans for the entire sheet metal design of the vehicle body, and distribute it to approximately 80 key individuals across the product development organization for review and comment before moving to detailed design of parts. This phase is remarkable because it seems to play a pivotal role in Toyota's process, yet it is altogether absent in US automotive processes. When coupled with Toyota's speed, productivity, and market and

financial success, it certainly seems kozokeikaku is worth a look.

What I believe Toyota is practising with kozokeikaku is a specific instance of a more general design practice that has received some, but perhaps not enough, attention in the design literature. It is a stage in the design process that falls (roughly speaking) after concept design and before detail design. Some authors place it in a separate phase (e.g., 'system-level design' in Ulrich and Eppinger [3], 'preliminary design' in Dym and Little [4], 'embodiment design' in Pahl and Beitz [5]), while others incorporate specific activities at the end of conceptual design or beginning of detail design phases that seem to encompass similar design tasks (e.g., the configuration design task in Ullman's product development phase [6] and the component design specification step in Pugh's detailed design phase [7]).

I call this phase 'system-level design,' borrowing Ulrich and Eppinger's term, and mean by it a fairly specific set of design tasks that include key elements from what other authors have separately called product architecture, product layout, and configuration design. I have come to define system-level design as: exploration of and decisions about what the components and subsystems are and what their function will be; the basic geometry of the different pieces and how they will be arranged, including location, orientation, and grouping; and how the pieces will connect or interface together and with the environment. System-level design bridges the gap between the highly abstract, concept design and the concrete, detail design phases.

The importance of these design decisions is difficult to measure in practice. This is because it is difficult to know 'what would have been' had another path been taken. For example, if designers (expert or otherwise) produce design solutions

* Accepted 18 January 2006.

without ever really spending much time on system-level design questions, how would we know if better solutions would have resulted if they had? And the reverse is also true. Our problem is compounded in that the real power of system-level design may manifest in multi-disciplinary team contexts, not individual design tasks, where interface issues are even more critical, which makes the problems of observation more difficult. However, if we could accumulate a large enough sample of projects, with a broad enough range of approaches, perhaps we could distinguish patterns that would lead to certain conclusions about the relative importance of system-level design issues.

We have collected process data from a sample of student mechanical engineering projects, and independently assessed the quality of the student teams' work. Through statistical analysis we have been able to associate the amount of effort on specific design activities to project outcomes. This paper presents an overview of this work, specifically focusing on system-level design issues. After defining 'system-level design' more fully, a summary of the research methods and analysis results to date on a sample of student projects, and discussion thereof is provided. The paper concludes with a possible design tool for system-level design that is to be tested for its efficacy.

SYSTEM-LEVEL DESIGN DEFINED

System-level design falls in the transition from concept to detail design. Numerous authors have proposed that certain activities be done or decisions made in order to make this transition as smoothly and as effectively as possible. Pahl and Beitz [5] identify the transition phase as an overall layout design (general arrangement and spatial compatibility), preliminary form design (component shapes and materials), the production process, and solutions to auxiliary functions. Embodiment design, as they call it, consists of a series of analysis-synthesis iterations to improve the layout for a given concept. This iteration is necessary due to the complexity and difficulty in foreseeing the consequences of change in a highly interrelated system.

Dym [8] also focuses on layout design, which he defines as deciding the general arrangement of components and assemblies with respect to spatial compatibility, in transitioning from concept to detailed design. Otto and Wood [9] expand on Pahl and Beitz's concept of embodiment design, suggesting embodiment design as a way to develop and understand alternative concepts in greater depth when it is not clear that one alternative concept is superior. This suggestion recognizes that a complete assessment of an idea is not always possible at a conceptual level. The transition phase is then used to test preliminary designs of a concept.

Ulrich and Eppinger [3] emphasize product

architecture decisions in the transition from concept design to detailed design. Specifically, they point out the need to identify the product's subsystems, define their functions, and decide the degree of modularity as determined by (1) the separation of function among the components/subsystems, and (2) the complexity of interface. In other words, how easily can a module be replaced or updated without impacting the rest of the system?

Pugh [7], interestingly, does not distinguish a transition phase. Rather, he begins the detailed design phase with a step called 'component design specification' (CDS). CDS involves defining the constraints for components to include not only functional performance parameters, but also interface and spatial constraints imposed by the system configuration. The interaction between the subsystems of the design should be considered as constraints imposed by those subsystems. Similarly, Ullman [6] does not define a distinct phase, but incorporates specific transition activities between concept design and component design at the beginning of a detailed design phase he calls the product development phase. These transition activities include configuration (or arrangement) of components and assemblies of components, keeping in mind spatial constraints, and defining interfaces between components that support their function.

Thus, the transition between concept level design and detailed level design seems to be characterized by exploration of and decisions about: what the components and subsystems are and what their function will be; how the different pieces will be arranged, including location, orientation, and grouping; and how the pieces will connect or interface with other pieces, the user, and the environment in which the product will function.

To illustrate what might be considered system-level design, consider the design of a parachute release mechanism for a military application that would release upon water entry (an example from our data set). One idea for activating the release mechanism was to use a pressurized CO₂ gas cylinder, activated by a lanyard pull, which would pressurize a line to throw a piston. To understand this idea more fully, the team conducted a rudimentary design study on how the CO₂ gas cylinder could be mounted relative to the lanyard assembly, and on the connectivity of the cylinder, pressure regulator, and piston. The design team performed a similar design study in parallel on a competing concept involving a mechanical clutch. After conducting the two configuration design studies, the team was able to determine with confidence that the idea they had originally favored based on their concept level work, the clutch, was inferior to the CO₂ cartridge idea in terms of cost, complexity, part count, and anticipated design challenges and unknowns. It appears that an hour or two of system-level design work helped the team make a decision

that potentially saved many hours of iteration in detailed design.

In contrast, another team in our sample designed a system to rapidly discharge fire retardant from a cargo plane. Obviously weight is a serious consideration in aircraft applications. The team considered weight in the concept selection phase, but then did not revisit this design requirement until well into detail design after all the structural elements, motors, gears, and drive shafts had been specified. When they discovered their design was grossly overweight, the team immediately engaged in a series of iterations to reduce the weight of their design, eventually succeeding after several weeks and over a hundred person-hours of effort. If this same exercise had been done at the system-level, most likely the weight objective could have been met in much less time.

From our journal data and observation, we see a strong tendency for student teams to assume a certain configuration in their conception of the solution concept, then proceed immediately to computer tools to conduct detailed part design or detailed analysis. Just how important is it for design teams to explicitly consider configuration as a design problem in its own right? The following sections provide data to help answer this question.

RESEARCH METHODS OVERVIEW

In 1999 we embarked on a study to better understand student design processes by collecting and characterizing design process data from student capstone projects, measuring the 'goodness' of the products of these projects, and modeling the data to see how the process parameters associate with the outcome measures. We collected process data from mechanical engineering capstone projects via design journals kept by the students [10]. We trained the students in journaling, then periodically evaluated the journals for thoroughness throughout the semester to increase the quality and quantity of data recorded. Students were required to put time and date stamps on all journal entries, which gave us a means of quantifying the processes. Journals were retained at the conclusion of the semester.

A subset of the projects was selected randomly from among those with usable journal records for coding. We developed a coding scheme that identified four categories of design activity (problem definition, idea generation, engineering analysis, and design refinement) and three design levels (concept, system, and detail) [11]. Of particular interest for this paper is system-level design—'defining subsystems for a particular concept, and defining their configuration and interfaces'—in contrast to concept-level design ('addressing a given problem or sub-problem with preliminary ideas, strategies, or approaches') and detail-level

design ('quantifying specific features required to realize a particular concept').

Each journal entry received an activity code and a design level code in order to distinguish, for example, concept-level problem definition from system-level problem definition, or concept-level analysis work from detail-level analysis. Table 1 summarizes the dual coding scheme, which will be used extensively in the next section. Time values for each code were estimated from the journal time stamps, and entered into an electronic database by date. Data were then aggregated to the project level to increase the reliability and representativeness of the data [12].

To measure the outcomes of the student projects, we developed a client satisfaction questionnaire and a design quality rubric [13]. The client satisfaction questionnaire measured the degree to which the project's client was satisfied with the final product. Two composite measures (1–5 scale) were summed into a 2 to 10 satisfaction score, with 10 being high. The client responses were collected via telephone to ensure a 100% response rate. During those discussions we learned that client satisfaction was relative to the client's expectations when they initially came into the project. Thus, to get a more objective comparison of the quality of student products relative to one another, we contracted four professional engineers to evaluate the final reports (which included engineering drawings and analyses) using a design quality rubric. The rubric asked the evaluator to score the project along five metrics on a scale of 1–7 (7 being high), which were averaged to obtain a quality score for the project. A minimum of two practising professionals evaluated each project.

To date we have coded journals from 19 projects, although the analyses reported in the next section were done at a point when 14 projects had been coded; work to incorporate the additional five projects is ongoing. Journals from the 14 projects represent over 5000 pages of documentation from 47 individual journals and thousands of hours of student design work.

RESULTS FROM MODELING AND ANALYSIS

To date, the modeling efforts have focused on total accumulated time for each of the 12 activity/design-level parameters, e.g., the number of

Table 1. Coding matrix

Design activities	Design levels		
	Concept (C)	System (S)	Detail (D)
Problem definition (PD)	C/PD	S/PD	D/PD
Idea generation (IG)	C/IG	S/IG	D/IG
Engineering analysis (EA)	C/EA	S/EA	D/EA
Design refinement (DR)	C/DR	S/DR	D/DR

person-hours spent on concept-level problem definition for the entire project. This section summarizes the results from three analyses on these data.

The first analysis used a step-wise reverse elimination technique to create two multiple linear regression models: one using client satisfaction score as the response variable, the other using design quality score [14]. The independent variables were the number of person-hours spent on each design-level/activity combination (from Table 1) over the course of the project. Both models show excellent fit as measured by R -squared, and interestingly, show little overlap in the statistically significant variables (see Table 2, Regression analysis). Other parameters were also included in the analysis, such as team size, effort level, amount of report writing, project management, and presentation preparation; but none of these improved the model or provided a better fit.

The second analysis involves a more sophisticated modeling technique: virtual design of experiments (VDOE) [15]. In this technique, two principal components neural network models were developed relating the 12 process variables to client satisfaction and design quality respectively. In this case, the process variables were expressed as a proportion of total design time (rather than raw hours, as was done in the previous analysis). Both neural network models had six principal components and one hidden neuron layer. Then, using the neural networks to predict project outcomes, two 2^{12-4} fractional factorial design of experiments were conducted, one on client satisfaction and the other on design quality, to see the effects of the different independent variables. The data were analyzed using analysis of variance (ANOVA). The results of the two ANOVA models were then used to determine the relative importance of the significant factors by dividing the slope of the variables versus the response variable by the absolute value of the lowest magnitude slope (D/DR in both cases). The relative slope estimates of the two models are displayed in Table 2 below.

The third analysis used a productivity measure as the response variable [16]. Productivity was calculated by averaging the client satisfaction and design quality scores of each project (seeing that students are, in theory, trying to optimize both), then dividing by the total number of design hours dedicated to the project. The independent variables were the number of person-hours spent at each design-level/activity, as in the first analysis. To analyze the data, we conducted a factor analysis on the independent variables resulting in four factors that explained 86% of the variance of the original variables. These factors were then fitted to the productivity scores using a linear regression model. One of the factors was insignificant and was removed. The remaining factors were significant at levels much lower than 1%. We then multiplied the vector of regression coefficients and the

matrix of factor loadings for these factors to obtain an estimate of the strength of association each of the original variables has with productivity. These productivity coefficients, as we termed them, are displayed in Table 2.

Each of these analyses takes a different look at the data, so we would expect to see differences in each despite being based on the same underlying data set. Several patterns seem particularly relevant to this article. First, the data from this sample seem to indicate that concept-level design work and detail-level design work are both important to a successful design project; but equally important is the transition from concept to detailed problem-solving. Activities at all three design levels are significant in all the models presented. Furthermore, activities at the system design level are predominately in the positive direction, and in each analysis save the client satisfaction regression model, *system-level design activities have the strongest positive associations with project outcomes*. Interesting, system-level engineering analysis (S/EA) seems to associate negatively with productivity, but positively with the quality measures in the VDOE analysis. This may indicate that S/EA activities do contribute to quality perhaps more so than other activities, but may not represent the most productive use of time or may quickly reach diminishing returns.

A second pattern of interest concerns problem definition work. Problem definition, as coded, includes understanding the client's needs as well as information gathering from outside the team. Such effort is often significant in these analyses and, with one exception, in the positive direction. System-level problem definition work shows significant positive association with both outcome measures in the VDOE models and is the second highest positive association with productivity. It does not, however, appear significant in the regression models.

The results concerning idea generation are also intriguing. While many authors on design advocate idea generation as a very positive activity for design, these results show a more mixed picture. Idea generation at the concept level is not significant in the regression model nor in the factor analysis of productivity; it is significant in the VDOE models, but positive for client satisfaction and negative for design quality. Idea generation at the system-level, on the other hand, where significant, is always positive across the models, and is the most positive contributor to productivity. Idea generation at the detailed level is either insignificant or negative across the analyses. A possible partial explanation for these somewhat counter-intuitive results is that we are dealing with student (a.k.a., novice) designers who have difficulty generating ideas of substance without much design experience.

Finally, we note that design refinement work (which includes small changes and adding detail to existing ideas, and most CAD work), where

Table 2. Summary of analyses results

Independent variables	Regression analysis		Relative slope estimates from VDOE		Factor analysis
	Client satisfaction model	Design quality model	Client satisfaction model	Design quality model	Productivity coefficient
Intercept	4.20**	1.90**			
Conceptual problem definition	0.09**		8.20	4.96	-1
Conceptual idea generation			8.16	-36.50	0
Conceptual engineering analysis	-0.11**		-4.09	†	4
Conceptual design refinement		-0.16**	-11.83	-48.97	-19
System problem definition			9.46	40.46	11
System idea generation		0.06*	†	31.61	31
System engineering analysis			21.06	114.51	-10
System design refinement		0.12**	-4.13	†	1
Detailed problem definition	0.03**		†	-14.82	5
Detailed idea generation			-7.71	†	-12
Detailed engineering analysis	0.02**	0.02**	-6.06	†	1
Detailed design refinement	-0.01**		-1.00	-1.00	0
R^2	0.96	0.91			
Standard error	0.38	0.37			
Degrees of freedom	8	9			
n	14	14			

* $p \leq 0.05$, ** $p \leq 0.01$.

† Insignificant at $p \leq 0.05$.

significant, is mostly negatively associated with project outcomes. The one exception is system-level design refinement, which is positive in the design quality regression model. These results are not terribly surprising as much time spent ‘tweaking’ or iterating is often due to problems in the design.

The results regarding system-level design activities are striking: not only do they appear as the most significant positive variables in four of the five models, they account for less than 10% of total design time in this data set. This suggests that the transition phase between concept and detail design is potentially very important, and could be a significant contributor to higher design team performance along multiple dimensions. The usefulness of this design phase seems to be the ability to reason about design solutions in greater depth than one typically can with high-level, general concepts, but yet not have to expend the time and resource required for detailed design. It enables greater depth of understanding of a solution concept while keeping flexibility for alternative subsolutions. We’ve also observed that designs often fail at the interfaces, so design effort that explicitly considers interfaces within designed systems seems prudent. It would seem that a tool to help designers with system-level design is potentially quite useful. Our current work focuses on developing and validating tools to aid with system-level design.

TOWARDS A SYSTEM-LEVEL DESIGN TOOL

In many ways, system-level design is simply treating the configuration and layout problem as a design problem unto itself. As such, any of the

design methods/tools advocated in the literature may be helpful with this important phase of mechanical systems design. We have identified one tool that seems particularly useful, morphological analysis or functional decomposition [5], and adapted it to the system-level design problem.

The proposed method is as follows. After the designer team has narrowed the set of concept alternatives to a manageable number (perhaps three or so), the team identifies, for each concept, the key functions that are critical for that concept to achieve the overall design objectives. The function set for each concept alternative should be unique, although there may be some overlap with that of other alternatives. If the function sets are not unique, then either the concept alternatives are essentially the same, or the team is working at too high a level.

Once the functions are identified for a concept, the team brainstorms options to accomplish each function (very similar to morphological analysis), and fills in the matrix shown in Fig. 1. Once the matrix is complete, the team begins to understand the configuration issues by identifying which options for one function exclude options for other functions (i.e., if option 1A is picked, option 2C is not feasible; these are termed ‘exclusions’), and identifying which options depend on others (i.e., if option 3B is desired, then option 1B is necessary; these are termed ‘dependencies’).

Once dependencies and exclusions within the sub-function options are identified and documented, the team generates a set of configuration alternatives for that concept by looking at what set of options optimizes each sub-function. This should result in several configuration options for each alternative. The team then studies the configurations with respect to two key questions: (1) how will the components/subassemblies of the

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:

Fig. 1. Morphological system design tool template.

configuration interface with others and the environment, and (2) how well does the configuration meet the overall design objectives or requirements?

Through this discussion and investigation, the team identifies a 'best' configuration for each concept alternative. With a good understanding of each alternative's configuration issues, the team is in a better informed position to make a wise alternative selection decision.

Currently we are conducting a controlled experiment with engineering students to measure the effectiveness of the design tool and associated system-level design procedure. We hope to determine whether design performance improves with the use of the tool, and how realistic it is for practical use.

SUMMARY AND IMPLICATIONS

In summary, the data from these ME student projects seem to indicate that system-level design activity distinguishes the better performing design teams within this sample; in other words, the transition from concept-level design to detail-level design work is critically important. It appears important enough to be treated as a mini-design problem unto itself. Design methods and tools designed to specifically aid this transition would

seem to be highly useful, yet few if any tools currently exist.

One of the challenges we see is that system-level work does seem to involve detailed investigation, but not on everything. Good system-level work seems to entail knowing which details (the 'vital few') are important at that stage of design and investigating those thoroughly, leaving the many, less significant details until later. Making this even more challenging is the fact that few if any tools exist to help with this; we are given general guidance on what's needed, then told to 'just do it.'

Future work includes expanding the analyses above to the full data set we currently have (19 projects) to confirm the results of this first set of analyses. Future work will also focus on the development and validation of tools to assist the transition from concept to detail design. I also hope to test these ideas in other engineering disciplines to test whether the transition from concept to detail design is equally important in other domains such as electrical systems design, transportation system design, structural design, or multi-disciplinary design. Another avenue of possible future work is to conduct a parallel study in industry to see if we observe similar effects.

Acknowledgements—This work was funded by National Science Foundation grant #REC-9984484. I would like to thank the ME404 instructors and students at Montana State University for their participation in this work.

REFERENCES

1. J. K. Liker, *The Toyota Way*, McGraw-Hill, New York, (2004).
2. D. K. Sobek, II, Principles that shape product development systems: A Toyota-Chrysler comparison, Ph.D. Dissertation, The University of Michigan, 1997.
3. K. Ulrich and S. Eppinger, *Product Design and Development*, McGraw-Hill, New York, (2000).
4. C. L. Dym and P. Little, *Engineering Design: A Project-Based Introduction*, John Wiley & Sons, New York, (2000).
5. G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*, Springer-Verlag, New York, (2001).
6. D. G. Ullman, *The Mechanical Design Process*, McGraw-Hill, New York, (2003).
7. S. Pugh, *Total Design*, Addison-Wesley Publishers Ltd., Cornwall, Great Britain, (1991).
8. C. L. Dym, *Engineering Design: A Synthesis of Views*, Cambridge University Press, New York, (1999).
9. K. N. Otto and K. L. Wood, *Product Design*, Prentice Hall, Upper Saddle River, NJ, (2001).
10. D. K. Sobek, II, Use of journals to evaluate student design processes, *American Society of Engineering Education Annual Conference*, Montreal, Canada, (2002).
11. D. K. Sobek, II, Preliminary findings from coding student design journals, *American Society of Engineering Education Annual Conference*, Montreal, Canada, (2002).
12. See Ref. 10 for more detail on the codes and coding protocol.
13. The actual instruments and details on their development and validation can be found in: D. K. Sobek, II, and V. K. Jain, Two instruments for assessing design outcomes of capstone projects, *American Society of Engineering Education Annual Conference*, Salt Lake City, UT, (2004).
14. S. Wilkening and D. K. Sobek, II, Relating design activity to quality of outcome: A regression analysis of student projects, *ASME Design Theory and Methodology Conference*, Salt Lake City, UT, (2004).
15. D. K. Sobek, II and V. K. Jain, The engineering problem-solving process: Good for students?, *American Society of Engineering Education Annual Conference*, Salt Lake City, UT, (2004).
16. R. Costa and D. K. Sobek, II, How process affects performance: An analysis of student design productivity, *ASME Design Theory and Methodology Conference*, Salt Lake City, UT, (2004).

Durward K. Sobek, II, is an Associate Professor of Mechanical and Industrial Engineering at Montana State University. He has Ph.D. and M.S. degrees in Industrial and Operations Engineering from the University of Michigan, and the A.B. degree in Engineering Sciences from Dartmouth College. Dr. Sobek's research focuses on creating highly effective delivery systems for products and services. Current areas of application include new product development, engineering design education, and health care. His work has been featured in numerous publications, including *Harvard Business Review*, *Sloan Management Review*, and *IEEE Transactions on Engineering Management*.