

Key Attributes of an EDS Design Environment Built for Optimization

Sjon Moore
Mentor Graphics Corp.

ABSTRACT

Traditional methods for optimizing an Electrical Distribution System have always been less than ideal. Engineers are typically left with unsophisticated and manual spreadsheet tools. Any detailed feedback on critical metrics such as cost or weight typically relies on another organization and requires week or months. The resulting EDS is inevitably suboptimized, not due to a lack of engineering ability, but rather the tools available. To ensure that the EDS design is optimized from the outset, the design environment itself must be enhanced. There are four key attributes to this design environment.

Automation is required to ensure that designs are synthesized rather than manually created. There are many benefits of this, but efficiency and repeatability are most important to this conversation. The ability to measure and assess the design must be available in the design, while the engineer is designing. The metrics used to measure must be easy to create and modify. There must be multiple ways to view the resulting data to aid the engineer's understanding. Measuring a single design is important, but the ability to compare alternatives, or even the same design through time is required. Proper assessment of options requires the ability to compare them. The measurements provided must become enriched and more accurate as the design matures. Estimations in the architecture space will become exact figures in the harness manufacturing space.

This paper will discuss these and flow implications in detail and use some examples from the Mentor Graphics Capital suite.

CITATION: Moore, S., "Key Attributes of an EDS Design Environment Built for Optimization," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 7(2):2014, doi:10.4271/2014-01-0318.

INTRODUCTION

To enable designers to innovate and implement true optimization of the Electrical Distribution System, new tools and methods of designing are needed. A single unified environment within which the designer can work is required. It must provide significant amounts of automation for design creation and optimization. It must allow the designer to analyze the design by asking simple questions and getting simple answers that can be acted upon. It must include powerful yet easy to understand and create measurements for assessing the design.

This paper will lay a foundation for why the tool must contain these characteristics then describe why traditional tools and methods fail to achieve this. It will then describe the key attributes outlined above. A case study will provide an example of just how such a tool could be utilized by using the Mentor

Graphics' Capital suite as an example. Lastly, there will be a brief discussion on process and how such a tool drives modifications to existing processes.

DESIGN FREEDOM VS. DESIGN KNOWLEDGE

Design freedom describes the flexibility with which the designer can modify a given design. As a design progresses through its lifecycle this freedom is reduced. In order to ensure production timing schedules are met, designs must be frozen in a timely manner so that downstream processes such as creation of tooling, modifications of production lines and logistics can all be organized. As the lifecycle progresses, fewer aspects of the design could potentially be modified without significant impact to ancillary designs with significant costs. To take this to its extreme, once a product is in the field, implementation of a

change on that specific product comes with both economic and brand image costs. Inevitably, the later in the lifecycle, the less ability the designer has to make modifications. [1]

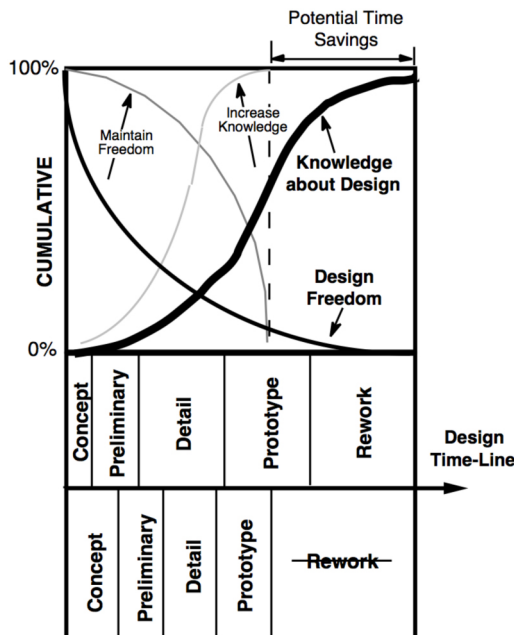


Figure 1. The Design Freedom and Design Knowledge curves against the typical design lifecycle. [1]

Design knowledge describes the theoretical and empirical understanding of the design and how it will operate in the real world. Early in the design lifecycle design knowledge is necessarily low, as many design decisions have yet to be made. The impacts of the decisions already made are also not yet fully understood. Knowledge increases as requirements and design parameters solidify. Any early simulation activities will further increase design knowledge, as behavior of the design is further understood. Once designs are frozen and prototypes are constructed, design knowledge again increases significantly with validation testing. Eventually it approaches its maximum once the product is in production and it is exposed to a vast array of environments and statistical factors start to come into play. [1]

Design optimization requires design knowledge to be effective. Specifically, the designer must know the extent to which the design meets or exceeds the requirements (is the product barely meeting the weight requirement, or is it significantly under), and they must be able to measure and assess key business drivers such as cost or profit. Once the designer is equipped with this information, they will be enabled to make more informed decisions and better balance the various attributes of the design.

The key is to ensure that the design knowledge is available as early as possible in the design lifecycle at a point in time when design freedom is at a maximum. It does little good to find optimization opportunities that cannot be implemented because the cost of scrapping tooling is too high. Instead, that

optimization opportunity should have been found earlier in the design cycle, before the design was frozen and long before tooling was constructed. Doing so enables the designer to implement more opportunities, and allows the organization to reap the rewards of the optimizations from the outset rather than trying to recover them once production begins.

FAILURE OF TRADITIONAL METHODS

The primary failure of traditional methods of optimization such as spreadsheets is that they do not fully enable the designer to increase design knowledge early enough. To overcome this, designers will use rules-of-thumb or simple estimations that are overly dependent on corporate memory and do not adequately reflect the reality of the design in front of them. The manual nature of these methods minimizes re-use, increases the possibility of error and can even allow for subjective bias to creep into the results.

Attempts to measure the performance of the design earlier in the process typically drive the need for simulation. This does indeed help move the design knowledge curve, but is still far from ideal. There is a significant amount of time taken for a designer to submit the task request and design information to a simulation expert, the time taken by the simulation expert to model, set up and run the simulation, then analyze the results into a meaningful answer for the designer.

To obtain accurate measurements of cost, the designer will typically rely on an external organization to perform detailed analysis using sophisticated models that provide very accurate results. The trade-off is that it takes a significant amount of time and the design must be essentially complete. Both of these make it impossible to get truly accurate costing knowledge until deep into the lifecycle at a point where the designer lacks the ability to affect meaningful change.

DESCRIPTION OF THE IDEAL ENVIRONMENT

Following is a description of the key attributes of an ideal design environment that enables a designer to fully optimize the design at a point in the lifecycle where such innovations can actually be implemented. These attributes enable this by moving the design knowledge curve to the left for maximum overlap with the design freedom curve. It does this by:

- Enabling the designer to garner knowledge and insight about the design at the exact point decisions are made
- Automating many tasks ensuring repeatable results with a minimal amount of time

Basic Attributes

This ideal environment has several basic requirements. It must be self-contained meaning that it is not required for the designer to ever leave the environment during the optimization process. Any optimization task such as analyzing performance

or measuring attributes must be launched and presented in the design environment. This can be in the form of graphical and tabular data reports, or even color-coded overlays on the design itself. This avoids the need for the designer to interrupt the thought process. The designer can stay focused on the optimization task rather than considering how to manipulate the tool. It also reinforces the relationship between the results and the design itself by displaying them in such close proximity.

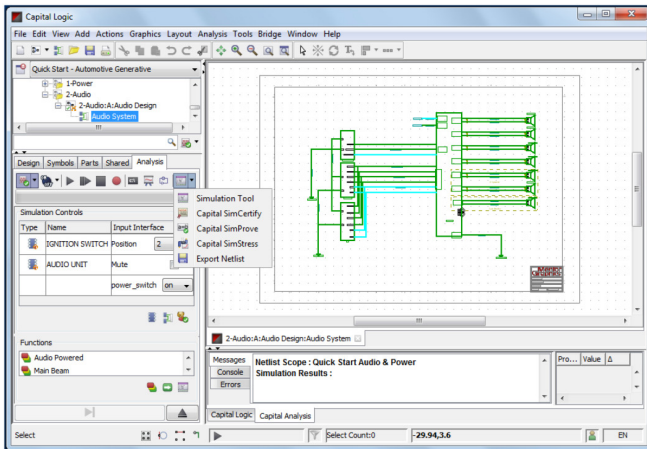


Figure 2. Analysis and simulation functionality embedded within the design environment. Color overlays provide behavior information. [2], [4]

Feedback from the tool must be dynamic and instantaneous. As an adjustment is made that impacts a measurement, the designer must be informed of it immediately. While analyzing the behavior of the design and interacting with the various switch settings and failure modes, the tool must immediately provide the designer with the results of that interaction. This provides the best cause-and-effect understanding for the designer. This wire size was changed (cause) and the cost went down (effect). By providing feedback dynamically and in the environment, the best possible user experience is achieved; one that fully assists the designer yet recedes into the background requiring no extra thought.

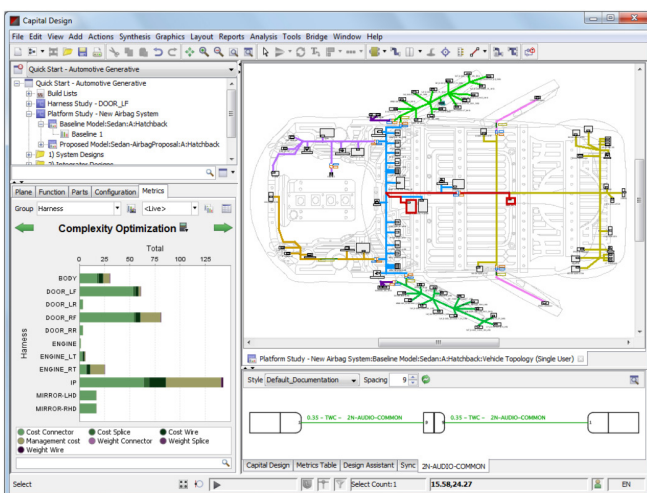


Figure 3. Metric feedback provided dynamically while the designer works. Design decisions result in immediate update of the metrics. [3], [5]

Ideally, these characteristics should be core to the tool itself, part of its DNA, rather than afterthought add-ons. Building these attributes into the foundation of the tools allows for the best possible user experience. Most importantly, it moves the acquisition of design knowledge to the earliest possible point, when the design decision is made.

Design Automation

Automating the design tasks increases the speed with which one can create complete wiring system designs for multiple alternatives. It also ensures repeatable results so that variation between alternatives is isolated to only the intended differences, and does not include manually induced variation. To fully automate the wiring design process one must abstract the thought and design processes to the platform level. Once this is accomplished it is simple to automate the synthesis of wiring designs.

A non-automated wiring design process will typically rely on the designer to intensely interact with the design and make many decisions as they go. Common decisions are whether to route the wire through a given harness and therefore adding an inline, or what size and insulation material type to use for a specific wire, or whether to use a splice or a multiterm (aka double crimp). This designer is not making these decisions at random, but is instead using some rules or guidelines that will direct the decision process.

Abstracting the design process allows the designer to instead think in terms of the rules and guidelines rather than their specific application. By capturing this information in a reusable format, it is then available for automation to perform the mundane tasks of applying them. This information is intellectual property and belongs to the organization. Ensuring that it exists in a re-usable format not only improves the design process, but also removes over-reliance on a small subset of highly skilled engineers.

The designer can also think in terms of the entire platform, rather than a single design at a time. Designing within the context of the entire platform ensures that the “big picture” is always in mind and minimizes the chances of mistakes falling through the crack. Interactions between devices or systems may not be obvious until they are placed within a real vehicle platform that is then subjected to real environments. These interactions are difficult to observe when designing across hundreds of sheets of wiring diagrams, but become obvious when designing the platform.

The platform design environment marries three types of data. The physical layout of the harness topology can be bridged from the 3D environment if it is available. Or, as is common very early in a product lifecycle, it can be easily drawn in the environment until such time as more accurate data is available. The logical system designs describe the devices and their logical connectivity. Lastly, the platform design environment contains all possible configurations. Therefore, marketing data in the

form of a product plan describes the variants and optional content and all of their relationships. This information combined will form the basis of the platform context and enable automation.

The captured rules can then be applied to the platform context and the wiring is created for the user. The decisions about which harness through which to route, what wire size and material to use, and whether to use a splice or a multiterm are made *for* the user rather than *by* the user. And these decisions are made with respect to all of the orderable configurations. What results is the synthesis of the entire wiring design. This convenient and re-usable way with sophisticated change management control. This allows this data to be confidently re-used to drive harness designs, service-ready documentation and so on. In this flow the wiring diagrams become an output rather than an input.

Synthesis of the wiring design through automation significantly increases the number of scenarios the designer can create and test. If the designer wishes to understand the impact of moving an inline to the opposite side of the vehicle, it is trivial to make the change in the platform design environment, then re-synthesize the wiring. Even changes to the rules and guidelines themselves that would normally be a difficult task to implement, is now a minor undertaking.

Automation also provides repeatability of the design output. If two designers are handed the same requirements and guidelines, they will likely produce 2 different wiring solutions. Automation avoids this entirely. Any variation between designs will be isolated to the driving differences such as the moved inline, or the changed guidelines. The variation will not be due to the fact that one designer developed one design while another did the alternative due to a vacation or illness. These non-value-add variations are noise and risk skewing the results of any optimization task.

In summary, design automation abstracts the design process and synthesizes wiring solutions. It does this to ensure consistent results that are an absolute requirement for effective optimization efforts. It also allows for a large number of alternatives to be considered very quickly. Combined, these two benefits move the design knowledge curve to the left to increase overlap with design freedom.

Analysis and Simulation

Simulation is a common method of acquiring design knowledge earlier in the lifecycle by simulating validation testing which typically occurs much later on physical prototypes, and it works well. But there are limitations imposed by currently available tools, processes and organizations that are limiting the amount of knowledge gained, and at what point in the lifecycle. For true optimization to occur, these limitations must be removed.

The tasks involved with setting up and running simulations tend to be very manual in nature and the tools involved are not designed with the typical design engineer in mind. The designer naturally has a set of relatively simple questions that need answering, such as whether or not the wires and fuses are sized correctly, or whether there are any high-risk failure modes that are not yet understood. Simulation tools are not designed to parse these types of questions to then do anything meaningful with them.

This drives a process in which the designer must pose their question to a simulation expert who is highly skilled in modeling the real world and in driving sophisticated simulation tools. The simulation expert must either get a translated copy of the design, or in some cases completely re-create it. The simple question asked by the designer drives the types of simulations that are run. The results that the simulation tools typically provide do not directly answer the designers question and therefore must be analyzed and correlated into a meaningful answer such as “yes” or “no”.

The tasks of taking the design and the question, compiling them into a set of simulation tasks, running the tasks, then compiling the results and providing them back in a meaningful way are all analysis tasks. They are time consuming when done in a manual process, and consume a highly skilled and highly valuable resource. They must be automated if true optimization is to be achieved. The ability to interact with the design, make design decisions, and then immediately see whether that design decision meets functional requirements, does not create any unexpected failure modes, or introduces any undersized components provides the designer with immediate knowledge of the behavior and validity of the design. The designer no longer must wait days or weeks to get an answer to their question. Automation provides answers immediately right at the point at which a decision is made.

Automation of analysis tasks increases design knowledge at the point of design decisions, but it can also automate optimization of component selection. This is most important for fuses and wires. To ensure that these components are sized to be robust enough, yet no larger, automating the sizing task is a significant time savings and ensures that component sizes are always optimal.

The tool must be capable of analyzing and simulating any platform design, determining the worst-case loads for various possible drive cycles and vehicle configurations, then make the appropriate recommendations. The tool must also allow the designer to instantly back propagate these recommendations right into the design, avoiding all of the negative consequences of manual interpretation and translation.

Automating the analysis tasks and embedding them into the design environment not only moves the acquisition of design knowledge to its earliest possible point, it also removes reliance on complex third-party simulation tools and dedicated experts. It provides the designer with meaningful answers