ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

UNCERTAINTY MODELING TO ENABLE SOFTWARE DEVELOPMENT PLATFORMS THAT CAN AUTOMATE COMPLEX MECHANICAL SYSTEMS DESIGN

Kevin OTTO (1), Christoffer LEVANDOWSKI (2), Anders FORSLUND (2), Hans JOHANNESSON (2), Rikard SÖDERBERG (2)
1: Singapore University of Technology and Design, Singapore; 2: Chalmers University of Technology, Sweden

ABSTRACT
Development platforms are automated software tools used to synthesize new designs. They are prevalent in the embedded system design domain, with applications ranging from integrated circuits, circuit boards, electro-mechanical controls, and entire networked systems. Historically, this has enabled rapid and error-free design of very complex embedded software and electronics hardware, even those that control mechanical systems such as aerospace and automotive controls through automation of the design process. The state of mechanical design automation has far less commercial adoption or industrial demonstration of development platforms in mechanical design. This paper elaborates on what challenges mechanical design automation faces to reach the level of design automation in the embeded systems domain. Given a design library approach, it is concluded that uncertainty management is a key issue for future research, including model uncertainty for mechanical design modeling. These issues are then contextualized using a case from the aerospace industy.

Keywords: design automation, uncertainty management, computational design synthesis, product modelling, simulation

Contact:
Christoffer Levandowski
Chalmers University of Technology
Department of Product and Production Development
Gothenburg
41296
Sweden
levandow@chalmers.se

1 INTRODUCTION

In this paper, *Development platforms* are discussed and explored in relation to the difficulties of mechanical system design. Development platforms are automated software tools used to synthesize new designs. They are prevalent as standard practice in the embedded system design domain, with applications ranging from integrated circuits, circuit boards, electro-mechanical controls, and entire networked systems. Design is executed through progressively more detailed analyses using the interlinked software tools of the development platform, from early software block diagram analysis to progress lower-level code development such as C-code and hardware in the loop simulation, to detailed chipset selection and board level logic design, and down to levels even more detailed such as custom integrated circuits. A key enabler is the semi-automated expansion of detail when progressing from one analysis to the next more detailed (e.g. Simulink to C). To do this, the integrated design tools make use of pre-defined libraries of standard components, where a component is represented appropriately in each design tool. The first key feature of development platforms is the hierarchical *shared library of components* and *functional requirements*, represented using a finite vocabulary upon which all possible system designs are computationally synthesized. The second key feature of development platforms is the *hierarchical progression* of ever more detailed models and analysis, effectively sequenced. The set of hierarchical software analyses along with the vertical library of components form what is called in this paper a *development platform*.

The process of using development platforms is a form of computationally assisted design referred to here as *Software-Platform Based-Design*, more specifically the computational process of system architectural matching between a library of pre-defined functional requirements to a set of pre-defined, quantified components, repeated progressively in layers. Historically developed in the embedded systems design community (Sangiovanni-Vincetelli 2007), software-platform-based-design has enabled rapid and error-free design of very complex embedded software and electronics hardware, even those that control mechanical systems such as aerospace and automotive controls. Vendors offer standard-work tools and processes as *tool chains* (Cadence 2012), as well as training courses on the standard work (*design flows*) to use the tools. There are also electronics manufacturer specific implementations, for example Intel (2010) or ST Microelectronics (Dubois et al 2006) offer design training and tools on their development platforms.

The state of mechanical design automation has far less commercial adoption or industrial demonstration of development platforms. There are no widespread commercial development platforms that allow the automated deep design of mechanical systems from functional descriptions. The state of research has been exploring this domain for some time, looking for significant design process productivity improvements. To do so however, this paper argues that several significant design science hurdles remain to be overcome. This paper elaborates on what challenges mechanical design automation faces to reach the level of design automation in the embedded systems domain.

2 MECHANICAL DESIGN AUTOMATION RESEARCH REVIEW

There is a rich history of research into mechanical design automation. Ward (1993) attempted early original work to develop a mechanical design compiler. Szykman et al (2001) worked for some time on a design library at NIST. Fogliatto et al (2012) provide a recent review of the literature in mass-customization design-automation. Among others, Helo et al (2010) and Ma et al (2008) have developed sales configurator tools, to assemble components to match requirements. Kortoglu et al (2009) speculate that a grammar parsing based approach would be effective for mapping functions to component libraries. Kortoglu et al (2010) offer a taxonomy of components for constructing mechanical design libraries for use in functional based synthesis. Wyatt et al (2009) compares available modeling languages and assesses their abilities in representing and modeling mechanical design concerns such as cost, weight, performance, reliability, etc. Melkote (2012) report on progress to create an automated fabrication facility (iFAB) where parts and assemblies can be fabricated based on assemblies of libraried mechanical components. It is not clear how well such resulting systems actually perform, nor how well even the assembly system itself performs. The *FANG design challenge* with the vehicle forge DARPA grantees (2012) seeks to automate the design of a military land vehicle using mechanical libraries and design automation.

3 DESIGN AUTOMATION USING HIERARCHICAL LAYERED DEVELOPMENT PLATFORMS

In early design phases, the design may be represented as simple hierarchical breakdowns of requirements and solutions, for example using function-means modelling. In detailed design, these hierarchical function-solution structures may be kept, however, to compute a system design solution using calculations. To do this, both the design *requirements* and the available *solution assets* must be computationally represented. Solution assets may be mechanical subassemblies or components in a system block diagram analysis, or could be functional blocks representing classes of solutions in an earlier functional synthesis step. At any level of abstraction, though, the design calculation is to match requirements to solution assets and eliminate infeasible solutions that do not meet the requirements. Software-platform-based-design sequences these calculations to use more simplified models earlier to filter large portions of the design space, but not over-filter.

Such solution-asset filtering calculations can be extremely complex, for example in the thermodynamic mechanical system domain this has been done using numerical search for worst case performance over operating conditions using simulation-code runs, all to simply evaluate one design configuration instance against a single thermodynamic performance requirement. Yet, it can be automated. Doing such calculations filters out solutions that do not meet requirements. Arguably, the successes mentioned previously have demonstrated this only in confined domains, e.g. sales configuration tools. This is not addressing the richness of the engineering design challenges faced in mechanical design, where progressively more detailed analyses are needed to ensure the system, modules, components and design features properly function and can be manufactured.

The software-platform-based-design approach addresses this sequence problem in the embedded system design domain. It is a unique methodology in that *it sequences the solution-set reduction calculations at increasing levels of abstraction*. There is not one model, but several models at increasing abstraction levels, where in each abstraction level, there are solution-asset models of the same solution assets (e.g., typically parallel component libraries in each tool). Progressively one eliminates assets inconsistent with additionally considered requirements, thereby refining the set of considered assets into smaller sets of consistent solutions. This is staged as progressively more detailed analyses, where such ordering is dependent on the problem domain. Therefore, platform based design is fundamentally consistent with the design theory of *set-based design* which has been argued as necessary for lean mechanical design, as executed in a non-computational manner by firms such as Toyota (Sobek et al 1999). Set-based design is achieved when bad portions of the design space are eliminated, rather than generating point solutions and the eliminating them. Depicting the entire design space is often prohibited by the mere size and the complex relationships (Deubzer and Lindemann, 2009). In spite of that, some authors, e.g. Raudberget (2011) propose methods for creating a comprehensive representation of a design space. Still, the paradigm of CAD and CAE today does not allow for modeling design spaces, as much as they are optimized for modeling point solutions. Burr et al. (2003) argues that IT-tools in general are not adapted to new design practice.

Platform-based design addresses these concerns by negotiating increasingly detailed analysis layers through *aligned models* of increasing detail; this is as advocated by for example Deubzer and Lindemann (2009). In the platform based design of aerospace thermodynamic systems, the heat exchangers, compressors and valves are typical components. There is a set of such components that are available to be used (possibly off the shelf or possibly not yet built). Each such component must be represented at each layer of abstraction, from a simple equation based architectural analysis layer to detailed sizing layer to a more detailed system dynamics layer, on down to detailed mechanical component feature design layers. This creates vertically aligned component libraries of progressive detail. At the highest level all heat exchangers have been modelled as a function which can be connected by pipes, with a few performance variables such as flow rate and heat rejection capacity alone defining the function. This can be used in architectural block diagram analyses to generate many architectural alternatives (many piping combinations of pumps, compressors, valves and heat exchangers). Several alternative architectures might all be deemed as meeting the functional requirements, at this level of rough analysis. Each of these architectures are then automatically refined in the next more detailed layer of analysis, through parameter alignment of and auto-configuration of component detail. In this example, more detailed thermodynamic sizing models of the heat exchangers form the next layer of analysis. These sizing models pre-exist and are available as parameterized component library elements in this more refined sizing analysis layer. The sizing analysis models are

physics based and are used to compute the dimensions of different heat exchanger configuration variables (lengths, widths, fin spacing, etc.) to provide the flow rate and heat rejection capability previously determined with the high abstraction level models. This refined analysis eliminates many configurations. This is continued into several more layers of detail, including geometric packaging, dynamic response, tolerances and other considerations as layered analyses. All such analysis layers have internal component library models of the same available assets (e.g. heat exchangers that can be designed and manufactured). Pre-configured physics based models are used at each layer to perform analyses that resolve issues of worst case operating conditions and testing requirements, variations of material and manufacture, and opportunities for design improvement. The result is a progress refinement of the design from many alternatives to an optimized mechanical design that meets all stated requirements at all levels of detail needed.

Though not the main focus in aero engine manufacture, powertrain development etc., and thus not in this paper, other research is exploring incorporating softer requirements such as styling in design automation (Reid, et al., 2012). There are also methods for combining optimization of manufacturing constraints with soft requirements such as economic and social sustainability (Hoffenson, et al., 2013).

Weber, et al. (2003) highlight the issue of the vast flora of models used to represent a mechanical design. To facilitate design automation, models need to be integrated and support the process of going from desired product properties to required product characteristics; design synthesis. In a later publication, Weber and Duebel (1993) claim that PDM and PLM system solutions exclusively manage characteristics data, and are unable to manage performance properties, a statement still true today. Modeling products in a way that allows for automatic design synthesis is immensely time consuming, and companies rarely have the competence to create those mathematical models of the design that can implement in a design automation IT-tool (Schotborgh, et al., 2009).

Others have argued that computational approaches are generally not possible for non-VLSI design domains, due to the inherent *higher power* nature of non-VLSI systems, the difficulty in creating *component libraries*, the *multi-physics nature* of mechanical design, and the *larger set of engineers* needed to design non-VLSI systems (Whitney 1996). Recent history with much increased mechanical design software tool capabilities are refuting these arguments, incorporating advances in computational mechanical component libraries, methods for fitting accurate and rapid reduced order models, and trade-space analysis tools. High power ESD applications have been demonstrated successful (Bathily et al 2012), as well as mechanical systems (Elmqvist et al 2003).

4 RESEARCH NEEDS IN MECHANICAL DEVELOPMENT PLATFORMS

Given these explanations do not seem to fully explain why electronics (and more generally embedded systems) can be more easily automated than mechanical systems, further scrutiny is needed. We observe that a key enabler of VLSI design was the reduction of uncertainty in field effects (heat, electro-magnetic radiation, etc.) to a tolerance (eg, minimum line widths and minimum spacing of circuit components to avoid interference and electrical leaks) (Nassif, 1984, Taylor and Boning, 2010). Such a reduction of difficult field effect analyses to geometric requirements simplifies the design task to a geometric constrained layout task which can be easily accommodated in VLSI design tools. No such equivalent mechanism to eliminate uncertainty exists in mechanical design. The multi-dimensional geometric performance evaluations inherently use physical simulation models. Unfortunately, such models cannot predict outcomes with sufficient accuracy to guarantee conformance to requirements, and the result is usually several prototype iterations. *Model uncertainty* management at multiple levels of abstraction of multiple interacting physics is needed to enable automated design flows at the different layers of abstraction, to ensure early computed decisions are consistent with later variations from models used in those early decisions.

Intrinsic to computational design automation is the analysis-based association of design components with functional requirements, through equations, simulations, relations or heuristic logic. To correctly refine a set of alternative designs computationally to a refined few, correct assessment of requirements violations or consistency must be determined. This is violated when the models used have inherent abstraction and therefore uncertainty. *Representing*, *forecasting*, and *pre-mitigating* model uncertainty is an essential design-science problem for mechanical design automation (and software platform based design in particular) to become successful.

A second design science issue with software-platform-based mechanical design is managing the further uncertainty surrounding the development of the manufacturing system. In electronic and

embedded systems design, the production system is given. In mechanical design, this is not so, the design of the tooling occurs simultaneously with the design of the components. The design of the assembly system occurs simultaneously with the layout design activity. While not a hard barrier, the issue remains that mechanical design cannot in general assume a given production system. The production system must also be auto-generated in the same manner with the product design.

A third design science issue with software-platform-based mechanical design is establishing effective process design flows. With the multitude of performance requirements, manufacturability and ease of assembly concerns, etc., it is not apparent a-priori which analysis to do in what sequence. It is not clear there is an overall best sequence of design tasks to cover an arbitrarily large class of mechanical design problems.

A fourth design science issue with software-platform-based mechanical design is the perceived limitations found in using pre-defined sets of quantified available components. The design space is restricted to the set of pre-defined alternatives. In design practice, this is not really a concern since the design engineers are always restricted to certified components, and new designs with large system design gaps can be assessed against critical components and new component targets thereby defined. Yet, a refined argument surrounds the inherent reduced design space one finds with a finite set of quantified available components (Doyle and Csete 2007). The infinitely large design space is reduced to a countable set of alternatives using a finite vocabulary quantifying the available components (the so-called *hourglass nature* of complex systems). The vocabulary chosen to quantify the components may not be the most effective descriptor variables for the system design problem.

5 UNCERTAINTY QUANTIFICATION

Over the past few decades, advances in computational capabilities have made computer simulation of physical processes an important tool used in the design of engineering systems (Agarwal, *et al.*, 2004, Oberkampf and Trucano, 2002). *Uncertainty quantification* is the scientific field of quantitative characterization and reduction of uncertainties in applications. As computer simulation modeling is a commonly used approach to study problems in uncertainty quantification, a framework has been developed to account for different types of uncertainties. Three different classes of uncertainties are identified (Agarwal, *et al.*, 2004, Apley, *et al.*, 2006, Oberkampf, *et al.*, 2002, Youn, *et al.*, 2007):

1. **Aleatory uncertainty** is also known as *irreducible, inherent or stochastic* uncertainty or *variability*. This uncertainty is associated with the inherent variation in the physical system or environment under consideration, for example, uncertainty of incoming material, initial part geometry, tooling setup, process setup, and operating environment (Chen, *et al.*, 2004). Aleatory uncertainty can generally be estimated by a probability or frequency distribution when sufficient information is available (Oberkampf, DeLand *et al.* 2002).
2. **Epistemic uncertainty** is also known as *reducible, subjective* or *cognitive* uncertainty. It can be defined as a potential inaccuracy associated with the deficiency in any phase or activity in the simulation process that originates in a *lack of system knowledge*, for example, uncertainty associated with the lack of knowledge in laws describing the behavior of the system under various conditions (Chen, *et al.*, 2004).
3. **Error** is defined as a *recognizable* inaccuracy in any phase or activity of modeling and simulation that is *not* due to a lack of knowledge. This error can be either *acknowledged* or *unacknowledged*. An example is the uncertainty associated with the limitations of numerical methods used to construct simulation models (Chen, *et al.*, 2004).

There is also a distinction being made between *acknowledged* and *unacknowledged* errors. Acknowledged errors are inaccuracies that are recognized by analysts, whereas unacknowledged errors are inaccuracies that are *not* recognized by analysts (Oberkampf, *et al.*, 2002). Oberkampf *et al.* (2002) have put forth a comprehensive framework for categorizing uncertainties in activities conducted during the phases of computational modeling and engineering The simulation process can be divided into six phases, each of which introduces its own set of uncertainties.

The *conceptual modeling of the physical phase* occurs before any mathematical or simulation models have been initiated. The initial step is *system/environment specification*—the process of determining which physical events should be considered and where to define the boundaries between system and environment. This phase introduces epistemic uncertainties. When modeling physical events with mathematical or simulation models, *scenario abstraction* is a prerequisite. For instance, dynamic

phenomena will often be simplified to quasi-static models. After systems, environments and scenarios have been specified, options for possible *physics couplings* should be identified. .

This *mathematical modeling* translates the conceptual model into detailed and precise mathematical problems, *i.e.* analytical statements of the problem. Even the most complex computer simulation is composed of many mathematical submodels. The mathematical modeling includes the complete specification of all the *partial differential equations* (PDEs), *auxiliary physical conditions*, *boundary conditions* (BCs) and *initial conditions* (ICs). Uncertainties in the PDEs can for instance be found in the conservation equations for mass, momentum and energy, which form the basis of CFD and FEA simulations. These uncertainties can be either epistemic or acknowledged errors. Examples of uncertainties in the auxiliary physical conditions might be limitations in turbulence models of CFD simulations, or in material-constitutive equations in FEA. Uncertainties in BCs and ICs include uncertainties in loads and geometries. *Nondeterministic representation* activities are associated with assigning probability distribution functions (PDFs) to uncertainty parameters.

In the *discretization and algorithm selection* phase addresses the conversion from continuous to discrete mathematics, which is usually needed to calculate a numerical solution. In this phase, all of the spatial and temporal differencing methods, discretized BCs, discretized geometric boundaries and grid generation methods are specified in analytical form. Algorithms are prescribed, but the spatial and temporal step sizes are not specified. Another activity is specifying the methodology that will be used to accommodate the nondeterministic aspect of the problem, such as Monte Carlo or response surface methods.

In the *computer programming* phase, software errors are introduced. *Input preparation* refers to how the mathematical model is converted into data elements usable by the software. Module design, coding, compilation and linkage refer to the construction of the software itself.

In the next phase the *numerical solutions* are computed. This phase includes uncertainties associated with *spatial*, *temporal* and *iterative convergence*. An example of spatial convergence is mesh density errors, whereas temporal convergence concerns time steps of dynamic simulations. Iterative convergence is usually an issue when working with CFD simulations.

The final phase deals with the *representation of the numerical solution*. Computer simulations generally have millions of data points and in order to represent these, post-processing is needed. A common tool is color-coded, three-dimensional graphical visualization. Another method would be to extract mean, minimum and maximum values of a simulation to draw conclusions. These activities are dependent on interpretation, and as such, may introduce unique types of error.

6 UNCERTANTIES IN TURBINE REAR STRUCTURE DESIGN

A case from the aerospace industry is used to contextualize uncertainties in reference to a real design case, in this case illustrated by a turbine rear structure (TRS) (Figure 1). The TRS, located in the rear end of a jet engine have a range of functional criteria from various fields of engineering. They need to be able to withstand significant thermal and structural loads. In addition, to optimize fuel efficiency, they need to be as light and aerodynamic as possible. These functionality criteria must be balanced in order to obtain an optimal design.

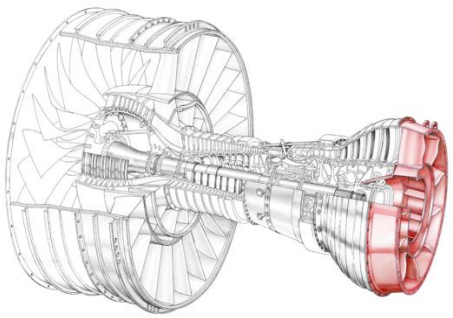

Figure 1: Turbine Rear Structure (marked in red)

However, manufacturability criteria are often difficult to quantitatively assess in the design optimization process. As a result, designs optimized from a functionality perspective are expensive or unfeasible to realize in practice. To avoid this scenario, functionality and manufacturability need to be balanced in order to find the truly optimal design (Runnemalm, *et al.*, 2009). One of the key

limitations of manufacturability is geometric variation, *i.e.* the dimensions of a manufactured product deviate from the nominal geometry. The structures are usually welded assemblies consisting of cast, wrought and sheet metal parts. The ingoing parts all have some degree of geometric variation. This part variation propagates through the fixturing and welding process into the final assembly and ultimately affects the performance of the engine. The assembly variation is dependent on part design, placement of fixturing points and welding sequence, and the impact of these late phase variations can be vastly different simply depending on the initial design concept and architecture.

A number of uncertainties need to be managed to be able to apply design automation to the turbine structure design process. On a conceptual level, there is epistemic uncertainty in interface loads, as well as uncertainty raised from using abstract models as compared to the later more detailed models. On a mathematical level, uncertainties might arise from the conservation equations for mass, momentum and energy, which form the basis for CFD and FEM simulations necessary to evaluate concepts. Also, auxiliary physical equations such as turbulence models in CFD simulations or material-constitutive equations in FE-analysis add to the total error and uncertainty of the model. In addition, there will be uncertainties in boundary conditions and initial conditions. The conversion from continuous to discrete mathematics comes with some error. From a numerical solution point of view, special convergence is somewhat addressed through evaluating different mesh densities.

The design of the Turbine Rear Structure is constrained by the interfaces to the high pressure turbine. The number of vanes connecting the shroud to the hub is one of the major factors in the design space. A large number of vanes create a strong and robust structure, whereas a small number of vanes results in less losses in aerodynamic energy. In other words, there is a trade-off to be made between aerodynamic efficiency, weight and structural integrity.

The practice for designing the structure is to set a constraint on the structural stiffness and thermal strain that the structure should withstand, and see how many vanes are needed to fulfil those criteria. It would be beneficial to have available a preliminary screening model consisting of simplified form equations, but that has been elusive. Both insufficient model accuracy of lumped parameter models and failure to consider manufacturing variability impact on aerodynamic performance have prevented that application. Instead, the first layer of software tools is a parametric CAD-model is connected to thermal and structural FEA software, is used to evaluate the design with respect to structural strength and thermal strain. Even so, the model is not perfect as it contains both aleatory and epistemic uncertainties. An example of aleatory uncertainty is that variations in assembly results in a non-nominal geometry. An example of epistemic uncertainty is the limited accuracy of the mesh used in the FEA simulations. The model is not perfect – instead it is estimated that the model deviates from reality with approximately ±5%.

In terms of mechanical design automation, several inherent difficulties become apparent. First, there are several alternative architectural layouts of the vanes that can be considered, in terms of number, orientation, and mounting. These must be determined early, and then subsequently analysed in lengthy engineering studies. Due to the inherent uncertainty in those analysis outcomes, it is very difficult to offer preliminary calculations that can be done at the architectural phase that can correctly assess one architecture as superior over the others. The result is a needed to iterate – explore several alternative architectures in detail using the more refined layer of analysis. This is the realm of finite element structural analysis, CFD analysis, and exploration of alternative fixturing methods to enable welding. Even with the detailed numerical studies, the results remain uncertain, again making the decisions contingent on the actual results of the subsequent phase.

The approach taken to resolve these uncertainties is parallel exploration of several alternatives, but this approach comes with a cost of that engineering work. Another approach is to incorporate *adjustment factors* into the design (Otto and Antonsson, 1993, Otto and Jacobson 2011), to allow the built prototype to be quickly modified into requirement compliance. In the engine example, it is relatively simple to change the flange size. This can be done during prototyping to bring the performance on target as predicted by the earlier phase simulation model analyses.

7 CONCLUSIONS

In summary, to enable mechanical design automation that can compute realistic and detailed design solutions as a chain of progressive interactive analyses and optimizations, several additional key design science questions include:

1. Given the recent history and successful one-off demonstrations in industry of software platform based design in mechanical systems, it remains unclear where gaps have arisen in practice and their root causes. Our experiences indicate model uncertainty is one key gap.
2. Given that any computed design configuration result is determined from physics based models with error, by what design flow should the hierarchical analyses be used to assess the risk and worthiness of refined modeling or prototyping of the computed configuration?
3. How can key uncertainties in mechanical system design automation be computed, and test conditions determined for prototypes based on this computed results?
4. Overall, how much of mechanical system and component design can be automated through a layered system of progressively detailed models of pre-configured mechanical components and their requirements (parallel libraries at various levels of abstraction)?

These questions can be well stated as statistical hypotheses and tested under industrial R&D conditions. This thereby provides for very interesting research opportunities in design science.

ACKNOWLEDGMENTS

This work was made possible through support from a research grant by the International Design Center at the Singapore University of Technology and Design. We would also like to acknowledge the Swedish Agency for Innovation Systems (VINNOVA) and the Swedish National Aeronautics Research Program (NFFP) for their financial support. This work was carried out at the Wingquist Laboratory VINN Excellence Centre within the Area of Advance – Production at Chalmers, also supported by VINNOVA. Finally we would like to thank ProViking research school for supporting this international collaboration.

REFERENCES

Agarwal, H., Renaud, J. E., Preston, E. L., and Padmanabhan, D., 2004, "Uncertainty Quantification Using Evidence Theory in Multidisciplinary Design Optimization", *Reliability Engineering & System Safety*, 85(1-3), pp. 281-294.

Apley, D. W., Liu, J., and Chen, W., 2006, "Understanding the Effects of Model Uncertainty in Robust Design With Computer Experiments", *Journal of Mechanical Design*, 128(4), pp. 945-958.

Bathily, M., Allard, B., Hasbani, F., Pinon, V. and J. Verdier, "Design Flow for High Switching Frequency and Large bandwidth Analog DC/DC Step-Down Converters for a Polar Transmitter," *IEEE Transactions on Power Electronics*, 27(2):838-847, 2012.

Cadence Corporation, *Allegro Design Reuse*, 2012.

Chen, W., Baghdasaryan, L., Buranathiti, T., and Cao, J., 2004, "Model Validation via Uncertainty Propagation and Data Transformations", *AIAA Journal*, 42(7), pp. 1406-1415.

Deubzer, F. and Lindemann, U. (2009) 'Networked Product Modelling - Use and Interaction of Product Models and Methods during Analysis and Synthesis', *ICED 09 - the 17th International Conference on Engineering Design*, August 24-27, Stanford, California, USA.

Doyle, J. and M. Csete, "Rules of Engagement," *Nature*, 446:860, 2007.

Dubois, F., Cano, J., Coppola, M., Flitch, J. and F. Petrot, *Spidergon STNoC Design Flow*, www.comcas.eu, 2006.

Elmqvist, H., Tummescheit, H. and M Otter. "Object-oriented modeling of thermo-fluid systems." *Proceedings of the 3rd International Modelica Conference*. 2003.

FANG Design Challenge, vehicleforge.org, 2012.

Fogliatto, F., da Silveira, G., and D. Borenstein, "The mass customization decade: An updated review of the literature," *Int. J. Production Economics*, 138:14–25, 2012.

Helo, P, Xu, Q., Kyllönon, S., and R. Jiao, "Integrated Vehicle Configuration System—Connecting the domains of mass customization," *Computers in Industry*, 61:44–52, 2010.

Intel Corporation, "The Tick Tock Beat of Micro-Processor Development at Intel," *Intel Technology Journal*, September 2010.

Hoffenson, S., Dagman, A. and Söderberg, R. (2013) 'Tolerance Specification Optimization for Economic and Ecological Sustainability', in M. Abramovici and R. Stark (ed.) *Smart Product Engineering*: Springer Berlin Heidelberg.

Ma, Y., Jiao, J. and Y. Deng, "Web Service-oriented Electronic Catalogs for Product Customization," *Concurrent Engineering*, 2008.

Kurtoglu, T. and M. Campbell, "Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping," *Journal of Engineering Design*, 20(1), 2009.

Kortoglu, T., Campbell, M., Arnold, C., Stone, R. and D. Mcadams, “A Component Taxonomy Framework for Component Design Synthesis,” *Journal of Computing and Information Sciences in Engineering*, Vol. 9, 2009.
Melkote, S., *Development of iFab (Instant Foundry Adaptive Through Bits) Manufacturing Process and Machine Library*, vehicleforge.org, 2012.
Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., and Alvin, K. F., 2002, "Error and Uncertainty in Modeling and Simulation", *Reliability Engineering & System Safety*, 75(3), pp. 333-357.
Nassif, SANI R., ANDRZEJ J. Strojwas, and STEPHEN W. Director. "FABRICS II: A statistically based IC fabrication process simulator."*Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 3.1 (1984): 40-46.
Oberkampf, W. L., and Trucano, T. G., 2002, "Verification and Validation in Computational Fluid Dynamics", *Progress in Aerospace Sciences*, 38(3), pp. 209-272.
Otto, K. and C. Jacobson, “Using Model Uncertainty to Reduce Verification and Validation In Noise and Vibration Problems,” DETC2012-70929, *ASME Design Engineering Technical Conferences*, 2012.
Otto, K., and E. Antonsson. Tuning Parameters in Engineering Design. *Journal of Mechanical Design*, 115(1):14-19, 1993.
Raudberget, D. (2011) 'Enabling Set-Based Concurrent Engineering in Traditional Product Development', *ICED '11 - the 18th International Conference on Engineering Design*, August 15-18, Copenhagen, Denmark.
Reid, T. N., Frischknecht, B. D. and Papalambros, P. Y. (2012) 'Perceptual Attributes in Product Design: Fuel Economy and Silhouette-Based Perceived Environmental Friendliness Tradeoffs in Automotive Vehicle Design', Journal of Mechanical Design, vol. 134, no. 4, pp.
Runnemalm, H., Tersing, H., and Isaksson, O., "Virtual Manufacturing of Light Weight Aero Engine Components", *Proceedings of ISABE 2009*.
Sangiovanni-Vincetelli, A., “Quo Vadis SLD? Reasoning About the Trends and Challenges of System Level Design,” *Proceedings of the IEEE*, 95(3):467-506, 2007.
Schotborgh, W., Kokkeler, F., Tragter, H. and van Houten, F. (2009) 'Why Is Design Automation Software Not Everywhere?', *ICED 09 - the 17th International Conference on Engineering Design*, August 24-27, Stanford, California, USA.
Sobek, D., Ward, A., and J. Liker, “Toyota’s Principles of Set Based Concurrent Engineering,” *Sloan Management Review*, Winter, pp 67-83, 1999.
Szkman, S., Fenves, S., Keirouz, W., and S. Shooter, “A foundation for interoperability in next generation product development systems,” Computer-Aided Design 33:545–559, 2001.Taylor, H. and Boning, D. (Year) 'Towards nanoimprint lithography-aware layout design checking', *Proceedings of Design for Manufacturability through Design-Process Integration*, April 02.
Ward, A., and W. Seering, “Quantitative Inference in a Mechanical Design Compiler,” *Journal of Mechanical Design*, 115(1):29-36, 1993.
Whitney, D., “Why Mechanical Design Cannot be Like VLSI Design,” *Research in Engineering Design*, 1996.
Weber, C. and Duebel, T. (1993) 'New Theory-Based Concepts for PDM and PLM', *ICED '03 - the 14th International Conference on Engineering Design*, August 19-21, Stockholm, Sweden.
Weber, C., Werner, H. and Deubel, T. (2003) 'A different view on Product Data Management/Product Life-Cycle Management and its future potentials', Journal of Engineering Design, vol. 14, no. 4, pp. 447-464.
Wyatt, D., Wynn, D., and P. Clarkson, “Comparing Representations for Product Architecture Design Through Life-Cycle Evaluation Methods,” *2nd Nordic Conference on Product Lifecycle Management, NORDPLM’09*, 2009.
Youn, B. D., Choi, K. K., Du, L., and Gorsich, D., 2007, "Integration of Possibility-based Optimization and Robust Design for Epistemic Uncertainty", *Journal of Mechanical Design*, 129(8), pp. 876-882.