Agile Manufacturing in 2020: Managing Complexity and Uncertainty in Product Realization

Jason Aughenbaugh
Systems Realization Laboratory
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Voice: 404-894-8169
Email: gtg224k@mail.gatech.edu

Advised by:
Christiaan J.J. Paredis
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0405
Voice: 404-894-5613
Fax: 404-894-9342
E-mail: chris.paredis@me.gatech.edu

Agile Manufacturing in 2020:
Managing Complexity and Uncertainty in Product Realization

Abstract

Designers face many challenges when designing products and systems. The marketplace of 2020 will be more global and faster paced than today's market. As the market expands and is shaped by new technology, a company's success will depend on how quickly and appropriately it can generate and apply knowledge to a specific problem. ADG's greatest asset is the knowledge of its employees, so this knowledge must be fostered and well managed. In order to design the complex, multidisciplinary systems that will dominate manufacturing in 2020, ADG must adopt a new approach to design that allows them to systematically decompose a complex design problem into simpler sub-problems. Systems engineering provides such a framework. In an iterative, hierarchical fashion systems are decomposed into subsystems and requirements are allocated to these subsystems based on estimates of their attributes. My primary recommendation in this report is that ADG adopt a systems engineering approach to product development and use modeling and simulation in specific roles that reduce uncertainty during the design process. Because there are currently limitations to this approach regarding the representation of uncertainty and model integration, I also identify avenues for future research that ADG should pursue jointly with academic researchers.

Note: I approached the report by imagining a specific company, Agile Design Group (ADG). As suggested in the competition description, I have focused on the product realization process.

1 Introductory Comments

The CEO of ADG (Agile Design Group) has stated the goal of positioning ADG to be a high-tech global manufacturing enterprise in the year 2020. In this report, I share my vision of how ADG can achieve this goal. I outline policies and methods that I believe will make ADG an agile and successful manufacturing enterprise in 2020.

Compared to today, the world of 2020 will be characterized both by more complex systems and a marketplace that requires greater agility. My approach to preparing ADG for this world is motivated by my view that in addition to the *product system* being manufactured, a company must recognize the *human system* that designs the product system. In order to be successful in 2020, ADG must successfully manage its human system, its product systems, and the interactions between the two systems.

I set the context of my recommendations by stating that your key to success at ADG is your employees and the knowledge that they can bring to bear on your projects. However, employees face several challenges to using knowledge effectively. First, they cannot consider all options and all consequences simultaneously because human rationality is inherently bounded [1]. Second, before it can be used, the information and knowledge has to be created. Third, in order to apply information and knowledge effectively, designers need to have access to it. This requires storing knowledge and communicating between disciplines and across thought-worlds [2, 3]. In order to be a manufacturing leader in 2020, ADG must overcome these challenges. ADG can overcome these challenges by adopting a new corporate philosophy, taking a new approach to product development, and embracing new technology.

I begin my report by briefly explaining my view of design and manufacturing in the year 2020, including the market and technologies of the product system. I then present the context of knowledge management in which I set my recommendations. In the next section I present my recommendation of systems engineering as the design methodology for 2020. I then explain how modeling and simulation support this approach. Finally, I discuss some limitations of this approach and avenues for future research.

2 Manufacturing in the year 2020

Manufacturing, like all industries, both adapts to and alters the world around it. For hundreds of years, Western Civilization expanded geographically by establishing colonies and opening new trade routes. More recently, new markets have been opening as a result of *shrinking* geography. Railroads and then airplanes sliced travel times for physical goods. The telegraph, then the telephone, and most recently the internet have enabled almost instant communication of increasing amount of data, information, and knowledge between nearly any two points on Earth. These new connections have changed the way companies do business and affected consumer demands. Modern consumers want more of what the world has to offer, sooner. Companies can fill these increased demands in part by automating manufacturing.

I expect the trends of shrinking geography, increased consumer demands, and automation to continue because they are all driven by new technologies. New technologies continue to revolutionize the way business and manufacturing are conducted. As I address in the following, in 2020 ADG will face a new *marketplace* shaped by new *technologies*, and new technologies will shape the *product system* development process.

2.1 The product marketplace in 2020

ADG's market in 2020 will be marked by several characteristics, such as the following.

- Global markets for high-tech products
    - Expanding existing markets
    - Opening new markets (e.g. China)
    - Developing markets (e.g. South America, Africa)
- Competition from other global and local firms
    - Local firms with better local knowledge of the market
    - Firms operating under different laws, ethics, and costs
- Consumers with rapidly evolving demands
    - Shorter product lifetimes
    - Variations in tastes and preferences in different regions

2.2 The state of technology in 2020

In 2020, both the product marketplace and the development process will be affected by new technologies. The following are some examples.

- Computer advances
    - Smaller devices: portability to new places and new uses
    - Faster processing: more complex graphics, searches, and simulations
    - More memory and storage: handle larger problems while capturing and storing more knowledge
- Autonomy: greater use of robotics and other autonomous entities
- Greater interconnectedness: within systems and across systems, databases, and disciplines
- Increased knowledge management
    - Natural language translation
    - Computer interpretable knowledge
    - Improved visualization tools
- More powerful modeling: explicit measures of uncertainty and capture of model context such as assumptions, applicable uses, and accuracy

The last two items—knowledge management and modeling—require the most additional research between now and 2020. These areas also provide the biggest opportunities for improving product development. In this report I focus on how modeling and simulation supports the design process by reducing uncertainty.

2.3 Product systems in 2020

During the past 25 years, systems have become more complex and are no longer dominated by one technology [4]. Products such as cars and toasters now contain extensive computer components that perform functions that were previously either non-existent or implemented with mechanical components. More recently, devices combining multiple functions have become smaller and less expensive, such as the merging of the

mobile phone, the PDA, and the digital camera into one device. Fifteen years ago these individual devices were not even individual consumer products, let alone integrated products. Looking ahead 15 years, ADG must be prepared both to drive and to respond to the even faster creation, evolution, and synthesis of new technologies and products.
In the high-tech product market of 2020 it will not be sufficient to offer just a handful of models for high-tech products. Customer desires and preferences will vary within and across ADG's various markets. To be successful in 2020, ADG must meet as many of these preferences as possible. Therefore, ADG must have solid marketing and customer research programs that identify and shape these demands. Then it will be up to ADG's designers to provide families of products that are customizable to customer subpopulations. In order to complete the design process successfully in 2020, ADG needs to adopt a new design method and embrace more modeling and simulation, as I address later in my report.

3 Knowledge in a manufacturing enterprise

I noted earlier that your key to success at ADG is the knowledge that they can bring to bear on your projects. My view of knowledge in an engineering firm is influenced by my experience working for the MITRE Corporation. In its role as a Federally Funded Research and Development Center (FFRDC), MITRE must collaborate with many public and private agencies and respond with agility to public needs. Therefore, knowledge management at MITRE today (see [5]) can provide guidance to ADG.

3.1 Product Lifecycle Management

I believe ADG can learn from Product Lifecycle Management (PLM). One definition of PLM is *a strategic business approach for the effective management and use of corporate intellectual capital*, where intellectual capital is *the sum of retained knowledge that an organization accumulates in the course of delivering its objectives* [6]. PLM normally focuses on *structural capital*, the things (such as processes, information systems, databases, patents, etc.) that are left behind when employees leave [7].

PLM is a bit of a buzzword that is currently tossed around in order to market many products, some of which are not truly PLM tools but rather applications that support individual, separate needs. However, it is really the interactions between these many applications that are important. In some sense, PLM is to a business what an operating system is to a computer. It is a series of functions and workflows that define how the pieces—different applications and entities—can communicate with each other and interact with storage, such as knowledge repositories and databases.

In this way, PLM is not just a set of disjoint tools such as a CAD or FEA packages. PLM is the heart of knowledge management. In order to get the most out of its structural capital, ADG must embrace PLM principles and direct PLM research down appropriate paths in order to support true knowledge management.

3.2 Managing Employee Knowledge

Another component of a company's intellectual capital is its *human capital*, defined as *that which is in the minds of individuals: knowledge, competencies, experience, know-how etc.* [7] In order to create value for ADG, it is not sufficient for the employees to *have* knowledge, nor is it sufficient to have *mechanisms* and tools for sharing it. The employees must also be *willing* to share their and apply their knowledge. I divide the flow of knowledge into four stages in which ADG must empower its workforce:

1. Get knowledge to engineers.
2. Unlock the knowledge in the workforce.
3. Provide means to communicate knowledge.
4. Encourage others to listen.

My goal in the following suggestions is to emphasize the importance of considering the human system at ADG. I am not an expert in human resource management, so I encourage ADG to hire employees or consultants in these areas who can develop or revise the following suggestions.

ADG can improve the flow of knowledge by providing opportunities for continuing education and professional development, such as computer based training and reimbursement for night classes. ADG can also encourage employees to create and attend internal seminars by creating charge codes for such activities. You can also unlock your workforce's knowledge by providing job security and new employee assessment techniques. When employees feel that they will be rewarded for sharing knowledge (rather than holding it in reserve in order to make themselves less dispensable), they will more readily share their knowledge and experience. Such assessments may prove elusive, but they are a research task for human resource experts.

In a globally distributed firm such as ADG, management can improve the exchange and application of information and knowledge by adopting a clear corporate philosophy. The corporate philosophy serves as the common thread that forms a cohesive bond between employees who are distributed over many countries, cultures, and languages. In order to establish this bond, ADG will need to vigorously recruit employees who believe in the same philosophy. One way to achieve this would be to increase the number of internships and co-ops available to university students and young professionals. Such short-term positions give the employees and ADG the mutual opportunity to discover if each is prepared to make a long-term commitment to the other.

4 A design method for 2020

The interface between a product system and the human system that builds it is the design method. One successful design method is *systematic design* [8]. Systematic design is based on several strong principles. The central of these principles is a belief in the need for a systematic approach to engineering design and decision making. Without a systematic approach, it is unlikely that designers will choose the correct path, and without the correct path, it is unlikely that the correct solution will be found. The main goal of an engineering design methodology is therefore to guide the designers down a path that is likely to succeed.

The core principles of systematic design will be relevant and valuable in 2020. However, systematic design has certain limitations.

- **Systematic design does not account for distributed design teams.** Systematic design proceeds in a linear manner, and assumes that there is only one design team working on the one design project. The process needs to be adapted to support concurrent processes carried out by distributed teams.

- **The process does not explicitly handle the types of uncertainty encountered in design.** When designers need to make decisions, they do not always have all the information that they would like to have. Systematic design does not address uncertainty in decisions.
- **Systematic design does not incorporate available computer resources, including modeling and simulation tools.** Systematic design was developed before the universal availability of computers in engineering design. The design process needs to be integrated with the computer tools of the present and future.

Given these limitations, ADG should move towards a new design process that embraces similar principles but at the same time better addresses the design issues and available technologies of 2020.

4.1 The need for systems engineering

The range of knowledge required for systems design in 2020 is beyond the experience of most engineers. Consequently, engineers must work in multidisciplinary teams of designers that rely on modeling and simulation in order to complete the design process. A holistic, hierarchical decomposition approach to the process of decomposing a system, delegating tasks to different design teams, and integrating separately designed subsystems is provided by s*ystems engineering [9, 10]*

Systems engineers recognize the interaction between the *product system* being designed and the *human system* that designs it. In this context, systems engineers coordinate and supervise the transition from stakeholder needs to the specifications that define the product. This approach allows multiple objectives to be considered at the systems level, including the consideration of product's entire lifecycle, as advocated in concurrent engineering [11].

Several models of the systems engineering product development process have been developed [9, 10]. The most commonly adopted model of the development process is the *Vee* model [12, 13] shown in Figure 1. The left side of the Vee represents the decomposition of the system into subsystems and the preliminary definition of those subsystems. The right side represents the integration of the fully detailed subsystems and system qualification. The base of the Vee is discipline design, the phase in which engineers fully define the system components.

The Vee model distinguishes between two types of engineers—systems engineers and discipline engineers. System engineers oversee and execute the process from a high-level, holistic perspective. The horizontal line dividing the top branches of the Vee from its base is the point at which systems engineers hand the specifications to the discipline engineers (i.e. mechanical, electrical, chemical, or computer engineers) who embody the physical system. Communication across this division significantly impacts the success of the system design process [3]. In addition to using well structured management and documentation, designers can bridge this gap using modeling and simulation.

Systems engineering and the Vee model are characterized by a hierarchical decomposition of the system. As shown in Figure 2, when designers make decisions at one level in the decomposition, they establish requirements for the next level. Each decision also affects attributes (such as mass or cost) that propagate back up the hierarchy. These attributes represent the factors that measure the extent to which a system meets its requirements. Designers usually cannot know these attributes exactly for each design alternative, but modeling and simulation of possible decisions helps designers to *estimate* them. With these estimates, designers can make better decisions as they decompose the system. These better decisions should result in a system that is more likely to meet its top-level requirements.

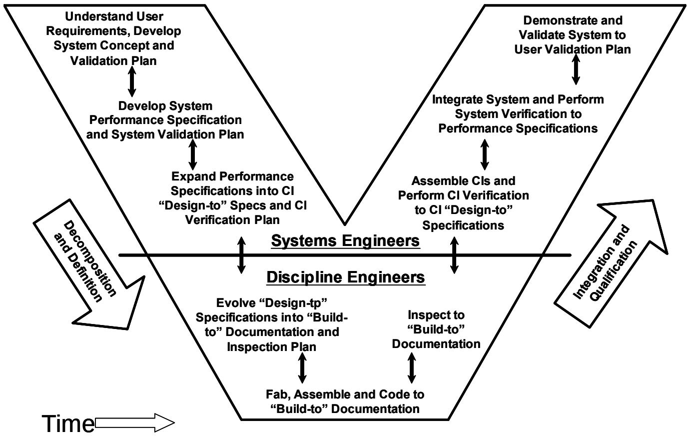

Figure 1: The Systems Engineering Vee [12, 13]

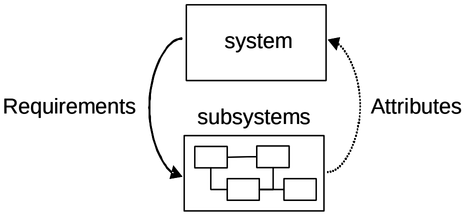

Figure 2: Requirements and Attributes

4.2 Limitations of the Decomposition Approach

In a perfect world, all systems could be decomposed into uncoupled subsystems and designers would have access to perfect information and knowledge. In such a world, systems design would be simple task. Subsystems could be designed independently and ADG's designers always would make perfect decisions. Unfortunately, ADG's designers face inherent challenges during the decomposition process. These include:

1. Design decisions are coupled and involve tradeoffs
2. Design decisions are made under uncertainty

Because of these challenges, designers often have to backtrack in the design process. Designers may have to revisit earlier decisions when new information is revealed, or when changes to one subsystem affect another subsystem. This iteration is undesirable because it costs time and money. The nature of these challenges is therefore important to improving the design process. In this section I focus on how designers at ADG can cope with uncertainty in the design process through the use of modeling and simulation.

4.2.1 Types of uncertainty

If designers do not understand the uncertainty involved in the information and knowledge that they are using to make a decision, they cannot make a rational decision. It is therefore useful for designers to recognize two basic types of uncertainty: *aleatory uncertainty* and *epistemic uncertainty* [14].

Aleatory uncertainty is a potential deviation from reality in a prediction or model due to natural random, stochastic behavior. Aleatory uncertainty is also known as variability, stochastic uncertainty, objective uncertainty, and irreducible uncertainty [15]. Because it results from inherent *randomness*, aleatory uncertainty can be represented using classical probability theory such as probability distribution functions.

Epistemic uncertainty relates to lack of knowledge [14, 15]. Epistemic uncertainty is also called imprecision [16], reducible uncertainty, or subjective uncertainty. It is generally incorrect to represent epistemic uncertainty using probability density functions. In some cases this is because there is not enough information to describe the relative likelihoods of events. In other cases, a probabilistic interpretation is altogether invalid. For example, epistemic uncertainty is introduced to a dynamics model by the decision to ignore frictional energy losses. An ignored frictional energy loss in general will not lead to a stochastic error. The deviation from reality is *systemic*, not stochastic, and therefore it should not be represented by a probability distribution. The model lacks knowledge about the true phenomenon (friction) and therefore contains epistemic uncertainty.

4.2.2 Epistemic uncertainty in engineering design

ADG's designers are faced with epistemic uncertainty in several ways other than modeling assumptions. These are:

- Lack of knowledge about the system's requirements
- Lack of knowledge about the system's environment
- Lack of knowledge about future design decisions
- Lack of knowledge about the system's emergent attributes

Theoretically, epistemic uncertainty can be reduced through sufficient study. For example, going to a more detailed model that includes more phenomena allows the designers to reduce the bounds on the uncertainty. The higher fidelity and more complete models also allow for errors in the higher level representation, such as missing inputs or outputs, to be identified and corrected. However, the entire idea of modeling is to abstract the essence of a system, not to reproduce the system in detail. At some point it is impractical to reduce uncertainty by further study or more detailed modeling because the costs will significantly outweigh the benefits.

Early in the design process, fully detailed modeling is actually impossible. During the system decomposition process, design knowledge is inherently incomplete because designers cannot know *a priori* what the final design will be [16]. There are still decisions to be taken, and designers naturally lack knowledge about what these will be. In the next section, I explain the ways in which modeling and simulation can be used to help cope with these four sources of epistemic uncertainty.

5 Modeling and simulation in systems design

In order to deal with epistemic uncertainty, I advocate the use of *executable specifications*. Originally developed for use in software engineering, an executable specification [17, 18] is a specification designed as a discrete event simulation model of the system's operational architecture as defined to that point in the design process. Executable specifications can be simulated directly, without humans needing to implement a separate discrete event simulation. This eliminates an extra problem of simulation model verification—the simulation model *is* the specification.

A key advantage of using executable specifications is that designers can explore the emergent behaviors before the physical embodiment of the system is defined. This means that designers can use them to explore design options while decomposing the system. The simulations reveal the operational behavior of the functional architecture as defined in the specifications. The designers can use this information to cope with epistemic uncertainty, discover the tradeoffs between coupled decisions, and to specify more appropriate designs and requirements for subsystems

5.1 Lack of knowledge about a system's requirements

Designers begin the design process without knowing the stakeholders' needs. Because these needs define the system's true requirements, designers lack knowledge about the system's requirements until they finish *understanding user requirements* (the first step in the Vee model). These requirements then guide the design towards a successful product.

The belief that products should be designed to reflect customer desires is shared by the management approach of quality function deployment (QFD). In this approach the house of quality serves as a conceptual map that provides a context for communication and planning [19]. According to Garvin, "quality means pleasing consumers" [20]. ADG's designers can complement QFD by interactively *exploring* the design specification with stakeholders using executable specifications.

An executable specification is a more detailed model of requirements than a requirements list. With this more detailed and executable model, designers can demonstrate to the stakeholders what they believe are the stakeholders' true needs. Through this process of conceptual and requirements validation [9], the designers can discover subtle conceptual errors [21]. The stakeholders can assess the specified operational behavior, decide if they are pleased with this behavior, and even clarify their true needs. Together with QFD, the use of executable specifications will help ADG's stakeholders and designers explore whether the design process is starting down a path that is likely to meet the stakeholders' needs.

5.2 Lack of knowledge about a system's environment

Designers usually begin the design process without full knowledge about a system's environment of operation. For example, before ADG builds and sells a high-tech product, no one uses that particular product. ADG hopes that after it launches a product, millions of people will use it. However, the exact way in which the product will be used is unknown. An executable specification helps designers explore the system's sensitivity to different use cases and environmental variables. Using formally designed experiments (such as those based on robust design principles [22]), ADG's designers can specify a system that will work well in variety of conditions.

5.3 Lack of knowledge about future decisions

Designers cannot know their future decisions *a priori,* so future decisions cannot be represented by probability distributions. For example, what is the probability that a particular cap screw will be used to fasten together two pieces that are not yet designed? Unless a particular cap screw is *required*, designers cannot answer this question.

Although early in the process designers don't know what the system will *look* like, they can know how it will *behave*. A model of the system's *specified* behavior will match the final product's *realized* behavior with probability near one because the product is *required* to match the model. Feedback control also helps to reduce operational uncertainty by making the realized behavior match the specified behavior. By using *executable* specifications of behavior, ADG's designers can reduce uncertainty about future decisions.

5.4 Lack of knowledge about emergent attributes

Designers can use executable specifications to explore the effects of the requirements and design decisions on the system's emergent behavior. The executable specifications include behavioral relationships that could be difficult to understand without simulation. For example, the simulation could help with the following roles:

- Reveal contradictions, timing problems, or deadlocks in the control logic
- Verify that the system as specified satisfies the nominal (intended) system behavior
- Explore the operational impact of the choices of subsystem requirements

These system characteristics emerge from the whole system and cannot be aggregated simply from the individual components. Designers can use the executable specification to explore how different values for design variables affect the operational behavior. This will not eliminate uncertainty, but it can help designers to estimate more accurately the effects that their decisions have on attributes.

6 Discussion

The systems engineering approach to design involves clarifying user requirements, decomposing and defining the system, completing detailed component designs, and integrating and qualifying the system. This process of bringing together the product system and the human systems dispersed knowledge can be supported by modeling and simulation. However, as more models are used in conceptual design, designers need both measures of confidence and systematic methods of application for their models.

Represent uncertainty. Researchers have explored several alternatives to classical probability theory, including possibility theory [23], fuzzy set theory [24], Dempster-Shafer evidence theory [25], probability-bounds analysis [26], interval analysis [27] and information gap decision theory [28]. ADG should sponsor and collaborate with academic researchers in these areas in order to relate these techniques specifically to systems design and model validation. Design requires computations involving both aleatory and epistemic uncertainty, so designers need mathematically sound methods to estimate and compute these.

Formalize model context. In addition to formally representing uncertainty in modeling, ADG should seek methods for formalizing a model's context and assumptions. Only with such representations of context, assumptions, and uncertainty can a designers form rational judgments of a model's accuracy and validity [29]. A solid representation of the information and knowledge in models would also facilitate model reuse and composable simulation [30]. Reuse of models also reuses the knowledge in those models. In this way, knowledge can be leveraged by the human system with little additional cost.

Integrate models: transformational design. Another area for joint research is in model integration. Currently, engineering models are in general not well integrated. There is a large gap between the types of models used in conceptual design and those used in the later phases of design. In software engineering, this gap is smaller, thus enabling the *transformational approach* to software engineering in which the executable specifications can be gradually replaced with the final code as portions of the final code are completed [17].

A research question for ADG to pursue is, “What are the fundaments for an analogous *transformational approach* to engineering design of physical systems?” Although the coupling between the model and implementation (code) in software is tight, the coupling between the models and products for physical systems is not as obvious. A formal representation of the information and knowledge in models at different levels of abstraction would enable designers to move smoothly between the types of models, or even use them simultaneously in a consistent, integrated manner. A formal representation of uncertainty would at least allow designers to compare the results of different types of models with an acceptable amount of confidence.

Impact on manufacturing. If research into these areas is successful, ADG could be the pioneer of a new approach to design. With a successful development of a transformational approach to design and formalized representations of model uncertainty, context, and assumptions, ADG will revolutionize the high-tech manufacturing of 2020 the way Henry Ford revolutionized the automotive industry. Combined with an new emphasis on the company’s human system, adoption of a systems engineering approach to development, and increased use of modeling and simulation, these revolutionary techniques will propel ADG to the forefront of manufacturing in 2020.

7 Summary

In order to effectively apply the corporation’s knowledge to product realization in 2020, ADG needs to adopt a systems engineering approach. This approach will enable ADG not only to handle the increasing complexity of products, but also to manage the increasingly distribute human system. Many of the challenges to system decomposition can be overcome within the systems engineering framework by using modeling and simulation. Modeling allows for a more accurate specification of the originating requirements, a more complete exploration of possible operational architectures, better analysis of emergent attributes, more accurate predictions of subsystem attributes, and design for robustness to various environments. Together these applications allow designers to make better tradeoffs in design decisions.

However, the current treatment of uncertainty in these models limits their usefulness. Designers must recognize and treat appropriately both aleatory and epistemic uncertainty. I expect that a more formal representation of uncertainty in design models could have a dramatic effect on the systems design and the engineering design field.

8 Acknowledgements

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional support is provided by the G.W. Woodruff School of Mechanical Engineering at Georgia Tech. I would like to thank my advisor Chris Paredis for his support and idea sharing. Finally, I'd like to thank the Systems Realization Laboratory for its collective intellectual support of my research and continue academic development.

9 References

[1] Simon, H., 1947, *Administrative Behavior*, The Macmillan Company, New York.

[2] Tversky, A., and Kahneman, D., 1974, "Judgment under Uncertainty: Heuristics and Biases.," *Science*, **185**, pp. 1124-1131.

[3] Cooper, L. P., 2003, "A Research Agenda to Reduce Risk in New Product Development through Knowledge Management: A Practitioner Perspective.," *Journal of Engineering and Technology Management*, **20**, pp. 117-140.

[4] Wiese, P. R., and John, P., 2003, *Engineering Design in the Multi-Discipline Era: A Systems Approach*, Professional Engineering Publishing Limited, London.

[5] Maybury, M. T., 2002, "Knowledge Management at the Mitre Corporation," MITRE Corporation, Bedford, MA, http://www.mitre.org/work/tech_papers/tech_papers_02/maybury_knowledge/maybury_km.pdf.

[6] 2002, "Understanding Product Lifecycle Management," PLM-11, Datamation Limited, Cambridge, UK

[7] Knowledgepoint, 2004, Knowledgepoint Online Glossary, http://www.knowledgepoint.com.au/starting_out/glossary.htm.

[8] Pahl, G., and Beitz, W., 1996, *Engineering Design: A Systematic Approach*, Springer Publishing, London.

[9] Buede, D. M., 2000, *The Engineering Design of Systems: Models and Methods*, John Wiley & Sons, Inc., New York.

[10] Blanchard, B. S., 2004, *Systems Engineering Management*, John Wiley and Sons, Inc., Hoboken, NJ.

[11] Hoffman, D. R., 1998, "Overview of Concurrent Engineering," *Proceedings of the 1998 Reliability and Maintainability Symposium, Jan 19-22 1998*, IEEE, Piscataway, NJ, USA, Anaheim, CA, USA, pp. 1-7.

[12] Forsberg, K., and Mooz, H., 1992, "The Relationship of Systems Engineering to the Project Cycle," *Engineering Management Journal*, **4**(3), pp. 36-43.

[13] Forsberg, K., Mooz, H., and Cotterman, H., 2000, *Visualizing Project Management: A Model for Business and Technical Success*, Wiley, New York.

[14] Parry, G. W., 1996, "The Characterization of Uncertainty in Probabilistic Risk Assessment of Complex Systems," *Reliability Engineering and System Safety*, **54**(2-3), pp. 119-126.

[15] 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 and System Safety*, **75**(3), pp. 333-357.

[16] Antonsson, E. K., and Otto, K. N., 1995, "Imprecision in Engineering Design," *Journal of Mechanical Design*, **117**, pp. 25-32.

[17] Gaskell, C., and Phillips, R., 1994, "Executable Specifications and CASE," *Software Engineering Journal*, **9**(4), pp. 174-182.

[18] Winton, I. A., 1994, "Multiple Domain Experience in Systems Engineering Using an Executable Specifications Tools," *14th Digital Avionics Systems Conference*, pp. 56-63.

[19] Hauser, J. R., and Clausings, D., 1988, "The House of Quality," *Harvard Business Review*, **66**(3), pp. 63-73.

[20] Garvin, D. A., 1987, "Competing on the Eight Dimenstions of Quality," *Harvard Business Review*, **65**(6), pp. 101-109.

[21] Harel, D., 1992, "Biting the Silver Bullet: Toward a Brighter Future for System Development," *Computer*, **25**(1), pp. 8-20.

[22] Taguchi, G., 1987, *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Cost*, UNIPUB/Kraus International Publications, Dearborn, Michigan.

[23] Dubois, D., 1988, *Possibility Theory*, Plenum Press, New York.

[24] Klir, G. J., 1997, *Fuzzy Set Theory: Foundations and Applications*, Prentice Hall, Upper Saddle River, NJ.

[25] Yager, R. R., Kacprzyk, J., and Fedrizzi, M., 1994, *Advances in the Dempster-Shafer Theory of Evidence*, John Wiley and Sons, New York.

[26] Ferson, S., 2000, "Probability Bounds Analysis Solves the Problem of Incomplete Specification in Probabilistic Risk and Safety Assessments," *Ninth Conference on Risk-Based Decisionmaking in Water Resources*, Santa Barbara CA, USA.

[27] Ferson, S., and Ginzburg, L., 1996, "Different Methods Are Needed to Propagate Ignorance and Variability," *Reliability Engineering and System Safety*, **54**(2-3).

[28] Ben-Haim, Y., 2001, *Information Gap Decision Theory*, Academic Press, London.

[29] Malak, R. J., Jr., and Paredis, C. J. J., 2004, "On Characterizing and Assessing the Validity of Behavior Models and Their Predictions," *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake City, pp. DETC2004-57452.

[30] Paredis, C. J. J., Diaz-Calderon, A., Sinha, R., and Khosla, P. K., 2001, "Composable Models for Simulation-Based Design," *Enginnering with Computers*, **17**, pp. 112-128.