8th Conference on Systems Engineering Research
March 17-19, 2010, Hoboken, NJ

Paper #nnn (For final camera ready papers use the ID number provided to you in your letter of acceptance)

Communication: Key Factor in Multidisciplinary System Design

G. Maarten Bonnema, P. Daniel Borches,
Rogier Kauw-A-Tjoe, Fred J.A.M. van Houten
Laboratory of Design, Production and Management, Faculty of Engineering Technology, University of Twente
P.O. Box 217, NL-7500 AE Enschede, The Netherlands
g.m.bonnema@utwente.nl

Abstract

Begin the paper with an abstract (50 to 150 words) that summarizes the topic and important results in the paper. **Include the abstract in the manuscript electronic format.**

Introduction

Multidisciplinary design is common practice these days. Most products are created in a joint effort of mechanical engineers, electrical engineers, software engineers and industrial designers. System Engineering is a set of techniques that helps to accomplish that cooperation, see Figure 1.

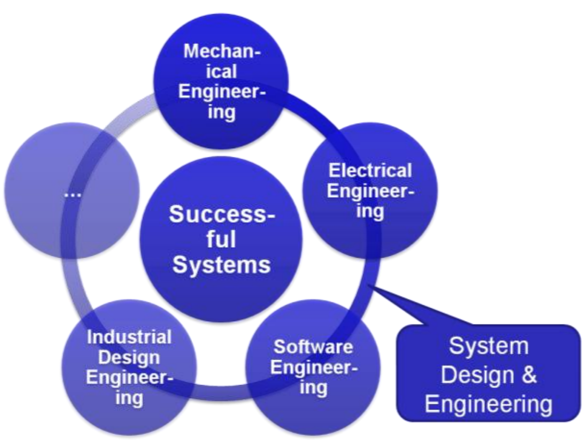

Figure 1: Multidisciplinary cooperation requires special attention.

However, the system engineering techniques alone do not provide the system design. Therefore, we will use the term “system design” to indicate the complete process of bringing to existence multidisciplinary systems.

System design is treated in literature. Several books deal with it ({INCOSE SEH Working Group}; Blanchard and Fabrycky 1998; Hinte and Tooren 2008; Maier and Rechtin 2000; Sage and Armstrong jr. 2000). These present a multitude of approaches and tools to facilitate system design. Founded on years of experience by the authors, they contain a wealth of information for the system designer. Also the present conference is a token of the relevance of the subject. Several interesting articles on the matter have been presented here (Martin and Davidz 2007; Muller 2009; Muller 2005) and elsewhere (Bonnema and Borches 2008; Borches and Bonnema 2008; Martin and Ferris 2008).

The goal of this paper is to show results from the System Design Group that provide hooks for further development and elaboration, and to define the group’s research focus. Moreover, a few interesting research questions are proposed, part of which can be treated by the System Design Group, part should be treated by others or in cooperation with others.

The group resorts under the faculty of Engineering Technology and the Laboratory of Design, Production and Management. The Laboratory of Design, Production and Management at the University of Twente originates from production technology research. Over the past decades the focus has shifted from this technology oriented research (the 70's and 80's), via product modelling (90's and 00's) to research on application, usability, concept design and systems design (00's). Central in this shift has been the fact that design gets more multidisciplinary and needs more focus on the ability to solve problems: moving from technology oriented to application oriented research. The System Design Group has emerged from the latter development.

In this paper, we will first look at several past and present projects performed at the System Design Group of the University of Twente. This relatively new group already has successfully completed several projects in close cooperation with industry. Based on these, and the running projects, the research theme for the group is derived. The links between this theme and the present projects is shown. Also directions for future research are indicated, including specific issues to be investigated.

The paper ends with several recommendations for system design research in general, and that of the System Design Group in particular.

Past and Current Projects

Selection of application cases. Defining projects where findings in the area of system design research are applied is not an easy issue. System design research aims at developing tools and methods for design of complex systems. Therefore, testing the applicability of the developed tools and methods requires complex cases. Figure 2 shows possible areas of application. They are classified using the scales *real* versus *laboratory* and *simple* versus *complex*. Note that it is practically impossible to develop a

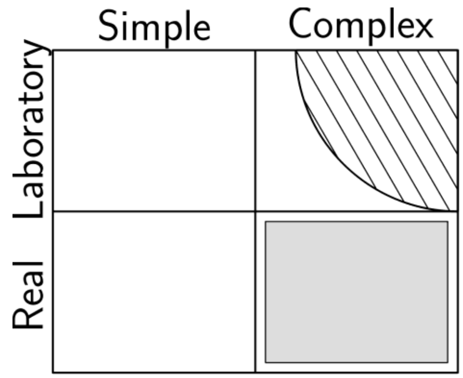

Figure 2: Grid to identify relevant System Design cases.

complex laboratory case, indicated by the hatching in Figure 2. It is clear that to practically verify the application of developed tools, we have to resort to real-life cases, indicated by the shading of the bottom-right corner of the grid.

For relevant cases, close cooperation with industry is thus vital, as also concluded by (Muller 2009) and the Embedded Systems Institute (www.esi.nl). Therefore, in the following project summaries, the industrial partner(s) is (are) mentioned. Also, for future projects, we will always involve a party from industry.

FunKey Architecting. It is found (Bonnema and van Houten 2006) that experienced system designers use a limited number of types of models, where the most important ones are:

1. System Budgets to divide system performance items over system's components (power budget, error budget etc.);
2. Analysis of physical behaviour;
3. Functional models like Functional Block Diagrams, and Function Structures;
4. Specifications and requirements.

In addition to the above, mathematical models are used to determine the system budgets, and sketches are used throughout the process for illustration of solutions and models in general. From this finding and general design literature, it is found that functions play a key role in the early phase of system design. However, functions alone are not enough. Connection to performance and the system decomposition is needed. For that, it is proposed to use key drivers to represent on the one hand the system's stakeholders' interest, and on the other hand the result of the designer's effort. Key drivers are generalised requiremenst. Examples are overlay for a wafer stepper, turn-around time for an aircraft, image quality for a medical imaging device. In general a system will have 5-10 key drivers to represent its performance.

Using a coupling matrix the effect of functions on key drivers is investigated, see Table 1, where a coupling matrix for a wafer stepper is shown. The key driver here is throughput. Other key drivers are:

- Critical dimension;
- Overlay;
- Cost per good die.

As seen from Table 1, nearly all top-level functions contribute to the throughput key driver.

The FunKey approach (from FUNctions and KEY drivers) provides direct clues for system improvement using a connection with the Theory of Inventive Problems Solving--TRIZ (Altshuller 1997; Bonnema 2006; Salamatov 1999).

Table 1: Functions-Key driver scheme for a wafer stepper. A cross indicates there is a contribution from the function to the key driver. Only one key driver is shown: throughput.

Functions	Key drivers
	Throughput
Load wafer	×
Prealign wafer	×
Wafer to expose chuck	×
Align wafer	×
Expose wafer	×
Maintain focus	
Position stage	×
Unload wafer	×

The approach has been applied in a new wafer stepper company: MAPPER Lithography (www.mapperlithography.com), and a company developing and producing waste balers: BOA systems (www.boarecycling.nl). In both cases, several interesting system concepts and system improvements have been found. Moreover, it provided a basis for system budgets (MAPPER) and architectures (BOA). Other results include increased insight in and overview over the system. Also, FunKey provides means to track technical progress and uncertainties. Overall, FunKey stimulates communication among the developers by making decisions explicit. See (Alink 2007; Bonnema 2008) for details.

Design for Evolvability/Darwin project. System requirements change over time; consequently, companies need to systematically evolve their products to cope with those changes. Since developing a system from scratch is time consuming and costly, new systems are often created by evolving an existing system. The knowledge that the company has about the system and the consequences of introducing changes determines its ability to effectively cope with system evolution.

Even in large companies, complex systems are typically poorly documented. The main architecture knowledge resides in the expert's minds, and only part of that knowledge is documented. Some key knowledge regarding the system architecture and design decisions may be lost, especially in long-lived systems, due to experts leaving the company, design decisions not documented and so on.

An MRI system, for example as developed by our industrial partner Philips Healthcare, requires a multidisciplinary design team with competences in areas such as mechanics, electronics, physics, material science, software and clinical science. Typically people are specialized in a single discipline, and each discipline uses its own vocabulary. Besides this, all the disciplines have to work together on different aspects of the design. Therefore effective communication across disciplines and departments is essential. The consequences of missing information or misunderstandings can cause serious problems and delays in the development process.

The result is an approach to **collect**, **abstract** and **present** architectural information in a fashion that can be understood and used by a broad set of stakeholders.

The main goal of an A3 architecture overview is to have a manageable architectural representation of a system aspect that enables system architects and designers to reason and communicate the consequences of system changes, see Figure 3. An architecture overview helps to provide a broad, comprehensive and easy to handle view of the system aspect under study.

The A3 Architecture Overview is a set of two sheets of A3 paper. It provides a model-based description of the system aspect, consisting of a functional view, a physical view and a quantification of key parameters view. Annotations of design constraints and design decisions are also present. The views are interlinked together by allocating functions into the physical view, pointers from views to other views, etc. The A3 Summary provides a compact text-based description to support the overview, structured for efficiency.

A3 Architecture Overviews can be used individually, or in group to deeper understanding, as the overviews provide references to other sources of information and other A3 Architecture Overviews.

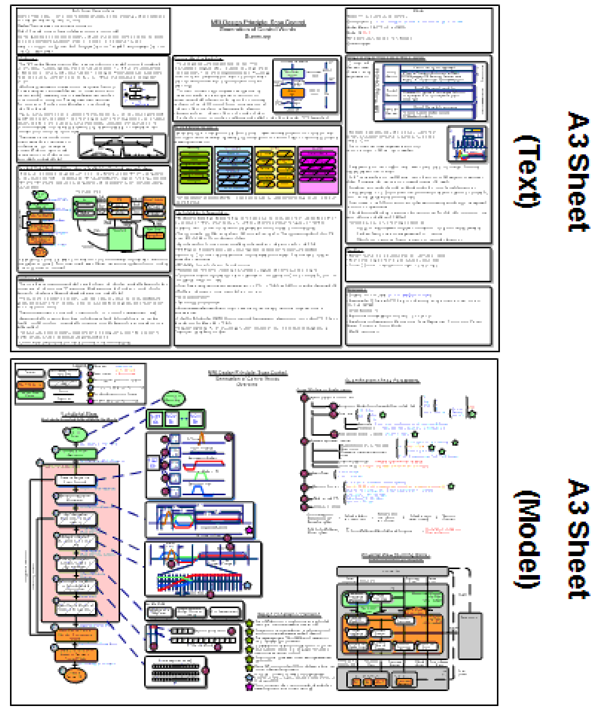

Figure 3: A3 Architecture Overview Example. Text intentionally made unreadable because of confidentiality.

Autonomous Litter-collecting Robot. Based on a third-year's student project, we have defined together with Stichting Nederland Schoon (Dutch foundation that focuses on a cleaner country) and Hako GmbH (a worldwide leader in outdoor cleaning equipment), a project to assist the cleaning personnel with a cleaning robot. The aim of the Cleaning Robot Project is to create an urban litter collection robot. Because of practical reasons, it is not possible to create the desired Cleaning Robot at once. To overcome these restrictions a roadmap is proposed which consists of four versions of the robot; two prototypes and two market ready versions.

All version have the same operation architecture (Figure 4). First the environment is recognised; obstacles and litter are identified. Based on the surroundings, a map is created by means of which the navigation path is determined. Navigation setpoints and the location of the litter is used to control the robot: motion control to drive towards the

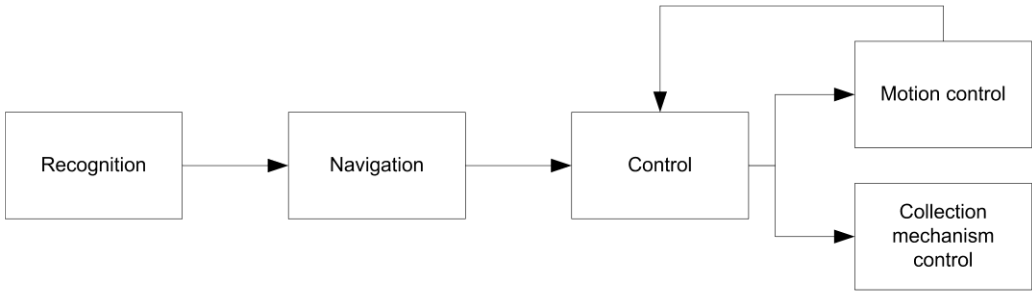

Figure 4: Basic operation architecture of the cleaning robot.

litter while avoiding obstacles and collection mechanism control to collect the litter in time.

The roadmap describes four versions:

- A functional prototype that presents the functions of the Cleaning Robot
- A Proof of Principle that presents the working principles
- The Cleaning Robot 2010 that works in a Semi-Autonomous Mode (SAM). The street cleaner operates (monitor and control) multiple robots.
- The Cleaning Robot 2020 that works in a Autonomous Mode (AM). The robots charge and empty the storage bin in a docking station and are monitored from a distance.

The system design of the Cleaning Robot is based on the Platform–Driven Development method (Halman et al. 2003). This consist of three system elements (modules, interfaces and standards) which allow a system to be easily divided into parts. This is necessary, since multiple students work on this robot: all need a clearly defined part from a system to design as a graduation or educational project.

FunKey (Bonnema 2008), also see Figure 5, is used to divide the system into coherent parts, called subsystem. By dividing the system into subsystems based on the relevant functions and stakeholders (key drivers [Muller, 2004], see Figure 5), each created subsystem has an added value, and a possibility to create a unique selling point to the system. However, because of practical reasons some subsystems created with FunKey are divided into smaller parts, adjusted to the level and available time of the students. The resulting subsystems will be called modules from now on.

The modules are described by means of functions (described in Functional Diagrams) and the interaction within the function. The modules are also described by means of the inter-modular interfaces in N^2 diagrams [INCOSE, 2000], and by the specifications of the system. The infrastructure in the robot is provided by mechanical, electrical and software frameworks: for instance energy supply for all modules by means of batteries.

After having created a system architecture, it is of great importance to ensure that all designers keep the architecture in mind at all times. Personal decisions instead of system

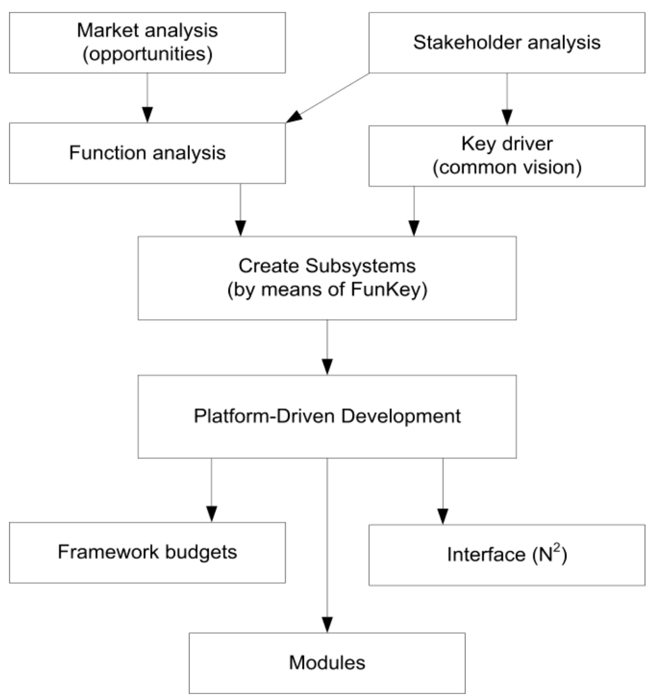

Figure 5: System design process in the cleaning robot project.

decisions can have great negative influence on the integration process of modules. Involvement by all team members was stimulated with weekly progress meetings with concise weekly reports showing decisions and open issues. This creates an awareness of other modules, even when these seem to be irrelevant for some of the team members. The involvement was created during several focused meetings:

- Integration meetings: all team members responsible for the frameworks prepare by making framework budgets and framework N^2 diagrams. All team members have to think about framework issues of their own modules, and therefore are forced to think about other modules.
- Interface meetings: people should know about the edges of their module. By involving the consequences for other modules in (interface) decisions, more thought-trough decisions are made. Functional N^2 diagrams make people aware of these interfaces and their implications.

By using a ranking method, the Trafic light evaluation method (Kauw-A-Tjoe 2009), team members are stimulated to formulate specifications more clearly.

Finally, during several team meetings, a clear focus has been put on identifying risks and defining appropriate risk mitigation scenarios ({INCOSE SEH Working Group}).

The next projects will be treated in less detail, because of space limitations.

Design Patterns in Mechatronics. This project is of a more fundamental nature. Nevertheless, it is clearly directed towards industrial application. The research will formulate a design architecture and a framework with which multi domain design processes can be integrated. It aims at the definition of an abstract model layer that connects the various domain specific models and design processes involved. This layer can, apart from integration, also be used to maintain model consistency and to automate design tasks. We hypothesize that patterns can be found in this abstract layer which can be reused in subsequent designs. These patterns will be called design patterns. They can be used as views from different domains on the same functionality. Each discipline sees its own familiar representation, while reusing information from other domains.

Industrial partners are vanderlande industries, océ and ASML. They provide a platform for application, but also input to the research itself.

TeleFLEX. The
Buzz-tracker.

Communication: the key factor

From the project descriptions given above, it is clear that in all projects a multidisciplinary team is involved in the design process. With present day products, there is almost always software involved. To run the software electronics are needed to drive the mechanics. Further, ergonomics, business and social sciences may play a role. The question is then how can all these developers be involved, and kept informed. Even more so as there are (should be) commercial people and managemers involved as well.

Common observations from all projects above are that to stimulate multidisciplinary cooperation it is required to:

- Have regular contact between specialists from different disciplines.
- Enable communication in a common “language”. This can be on the system level as shown in the FunKey project, the technology and product family level, as seen in the Darwin project, or at more detail levels, as shown in the litter collecting robot project.

- Have an appointed system engineer/system designer (or a team thereof).
- Focus on integration as early as possible. In particular the stepwise integration of two or more disciplines has to be aimed at.
- Involve the hardware as soon a possible; avoid prolonged simulation and optimisation as that may improve potential performance, but does not guarantee basic operation.

Although these observations are not all new, the crucial role of communication requires more attention in future research. In the next section, these observations will be translated in research themes for the system design group.

System Design Group Research Themes

The overall goal for system design research in general should be to support and assist the system designer in his/her work. Thereby focus should be on the system design tasks that are:

- Tedious
- Difficult
- Repeating

It is not wise to take the interesting and creative tasks away from the designer. These provide job satisfaction and motivation. (Csikszentmihayli 1990) describes the state of "flow" when people are challenged enough to avoid boredom and not enough as to create anxiety. This state of flow is related to the skills of the engineers. Moreover, it is questionable whether support tools will ever be able to provide creative solutions.

Communication is at the core of multidisciplinary cooperation and system design, as seen above. This communication should support both inside-out and outside-in communication. Here, inside-out communication is from the technology to the application of the system under design (SUD). Thus, what opportunities does the result of the engineer's effort provide the system buyer?

Outside-in communication is the other way around. What wishes, demands and requirements does the system buyer pose on to the technology?

A proper matching of these two communication streams will result in more focused system design processes, and avoid engineers aiming for perfect solutions, where a good solution will do. The other way around, the buyer should be aware of technological barriers, risks and limitations, because then the developers and the buyer can work together towards a good solution that is on time and not too expensive.

An example (Hinte and Tooren 2008 p.87) is that Airbus salespeople had promised to customers the ability to alter the wiring up to a very short moment before delivery. This, of course, is impossible for such a complex and interlinked system as an aircraft. If the salespeople would have been more aware of the limitations technology poses via inside-out communication, this would not have happened.

This example also illustrates the fact that communication should not only be stimulated among engineers of different disciplines, but also among engineers and salespeople, engineers and management, etc.

Therefore, we define the following research themes:

- Create High-level models: Creating a simple to use "language" that is understood by all disciplines involved. The language should be able to convey customer interests, technical opportunities and limitations, and result in simpler models that can be used by the more monodisciplinary oriented designers.
- Combine model types: As each discipline has its own set of frequently used models, it is necessary to investigate a way of connecting these models. Goal is that each discipline can look at its own models but

use data from other models where needed, without noticing.

- Condense information: We have observed that in contrast to the general idea, expert designers do not use models that are as complete as possible. They use models that are as simple as possible ("but not simpler", to paraphrase Einstein). The issue is to find the essence of the problem, and describe that as compactly as possible. The process of simplifying the model of the problem is very useful in finding the parameters and processes that determine the actually achieved performance.

In these themes, it is essential to understand the fact that a model is a limited abstraction of reality. Even more so, every observer will have a different view on the system, resulting in different conceptualizations, as shown nicely in (Martin and Ferris 2008). The other way around, when these different conceptualizations are combined, the model will be more complete. Thus, it is essential to make state-transition diagrams and functional block diagrams and power budgets and mechanical sketches and ergonomics mock-ups etc. Together they will provide a more realistic image of the SUD. Relating the different conceptualizations is an issue treated in the second theme.

The first theme tries to create a way to provide the system designer with overview, and the detail designer with context information (Bonnema 2008). Further, the "language" should be understandable by the customers as well. That will enable a constant flow of information from the customer to the (system) engineers (outside-in), and vice versa (inside-out).

The last theme aims at avoiding having to read through thick documents, finding inconsistencies and errors. The information should be presented in a concise manner, so that the essence is clear. Correctness should not be corrupted, though.

In Table 2 the projects are related to the themes defined. It is shown that most projects are connected to two themes. There is a focus on one of the teams in the FunKey, Darwin and Design Patterns projects. The TeleFLEX and Litter Robot project use the findings of the other projects and thus act as application cases. The Buzz tracker is different, as it was a single master student project. Nevertheless, the findings can be used in future research.

Table 2: Matrix relating the research themes to the projects treated.

Project	High-level models	Combine models	Condense information
FunKey	××		×
Darwin		×	××
Design patterns	×	××	×
TeleFLEX	×	×	×
Robot	×	×	×
Buzz tracker			×

Future

As the group already has good contacts with relevant industry in the Netherlands, the basis for research and evaluation is promising. Nevertheless, the number and intensity of partner industries should be improved; both nationally and internationally. The focus will be on Dutch cooperation, though.

As the themes have now been defined, it will be possible to deepen the research, and to maintain close relations between the projects. This stimulates academic discussions among the researchers. It will also be possible to have more bachelor and master students doing specific researches in companies. Even more, because of the contacts with different industries, a PhD researcher can have a master student apply his research in a different company than the one the PhD researcher is working for. This will improve the quality of validation (Martin and Davidz 2007).

Finally, the research results should be used to improve education of mechanical and industrial design engineers. Maybe even for civil engineers as well. We will work with people from the Electronics, Math and Computer Science faculty of our university as well, so courses can be improved for those students as well.

In this area, it is interesting to note that systems engineering is not formally part of the bachelor program for mechanical engineering, whereas it is part of the program for industrial design engineering.

It is our aim to have a basic course on systems engineering for industrial design, mechanical and civil engineering within three years. In the master program further deepening and widening will be aimed at.

Acknowledgements

The work presented under "**Design for Evolvability/Darwin project**" has been carried out as a part of the DARWIN project at/with Philips Healthcare Nederland under the responsibilities of the Embedded Systems Institute. This project is partially supported by the Dutch Ministry of Economic Affairs under the BSIK program.

References

{INCOSE SEH Working Group}. (2008). *Systems Engineering Handbook*, 3.1 Ed., INCOSE.

Alink, W. (2007). "Systematic, conceptual re-design of a waste baler," Master thesis, University of Twente, Enschede.

Altshuller, G. S. (1997). *40 Principles - TRIZ Keys to Technical Innovation*, Technical Innovation Center, Worcester, MA.

Blanchard, B. S., and Fabrycky, W. J. (1998). *Systems Engineering and Analysis*, Third Ed., Prentice Hall, Upper Saddle River, New Jersey.

Bonnema, G. M. (2006). "TRIZ for Systems Architecting." In: *TRIZ Future 2006*, Kortrijk, Belgium, I: 87--92.

Bonnema, G. M. (2008). "FunKey Architecting - An Integrated Approach to System Architecting Using Functions, Key Drivers and System Budgets," Ph.D.-thesis, University of Twente, Enschede, The Netherlands.

Bonnema, G. M., and Borches, P. D. (2008). "Design with Overview – how to survive in complex organisations." In: *18th Annual International Symposium of INCOSE (IS2008)*, INCOSE, Utrecht, The Netherlands.

Bonnema, G. M., and van Houten, F. J. A. M. (2006). "Use of Models in Conceptual Design." *Journal of Engineering Design*, 17(6), 549--562.

Borches, P. D., and Bonnema, G. M. (2008). "'Living' Architecture Overviews - Supporting the Design of Evolutionary Complex Systems." In: *CIRP Design Conference 2008*, CIRP, Enschede, The Netherlands.

Csikszentmihayli, M. (1990). *Flow: The Psychology of Optimal Experience*, Harper & Row, New York.

Halman, J. I. M., Hofer, A. P., and Vuren van, W. (2003). "Platform-Driven Development of Product Families: Linking Theory with Practice." *Journal of Product Innovation Management*, 20(2), 149-162.

Hinte, E. v., and Tooren, M. v. (2008). *First Read This*, 010 Publishers, Rotterdam.

Kauw-A-Tjoe, R. (2009). "The Cleaning Robot Project," MSc.-thesis, Twente University, Enschede.

Maier, M. W., and Rechtin, E. (2000). *The art of systems architecting*, 2nd Ed., CRC Press, Boca Raton.

Martin, J. N., and Davidz, H. L. (2007). "Systems Engineering Case Study Development." In: *5th Annual Conference on Systems Engineering Research 2007*

(CSER2007), Stevens Institute of Technology, Hoboken, New Jersey.

Martin, J. N., and Ferris, T. L. J. (2008). "On the Various Conceptualizations of Systems and Their Impact on the Practice of Systems Engineering." In: *IS2008*, INCOSE, Utrecht, the Netherlands.

Muller, G. (2009). "Creating an Academic Systems Engineering Group; Bootstrapping Systems Engineering Research." In: *7th Annual Conference on Systems Engineering Research 2009 (CSER2009)*, Loughborough, UK.

Muller, G. J. (2005). "Do Useful Multi-Domain Methods Exist?" In: *3rd Conference on Systems Engineering Research (CSER2005)*, Hoboken, NJ, USA.

Sage, A. P., and Armstrong jr., J. E. (2000). *Introduction to System Engineering*, John Wiley and Sons, Inc., New York.

Salamatov, Y. P. (1999). *TRIZ: the right solution at the right time: a guide to innovative problem solving*, Insytec, Hattem, The Netherlands.

Biography

Include a short biographical sketch (50 to 100 words) for each author at the end of the manuscript.