A Theory for the Systems Engineering Process

Louwrence D Erasmus
Graduate School of Technology Management
University of Pretoria
Pretoria, 0002
South Africa
Email: louwrence.erasmus@up.ac.za

Gerd Doeben-Henisch
Department of Intelligent Systems
University of Applied Sciences
Nibelungenplatz 1, 60318 Frankfurt am Main
Germany
Email: doeben@fb2.fh-frankfurt.de

***Abstract*—A theory for describing the systems engineering process using formal mathematical structures is presented in this paper. This abstraction of the systems engineering process makes it possible to concentrate on the operations and structures involved in the process without the distraction of the narrative word. An important aspect in the formulation of this theory is the inclusion of people as part of it. Further development of the theory will lead to the implementation of the mathematical description in simulation software to study the dynamic characteristics of and interaction of people with the systems engineering process as well as systematically validating the theory through empirical studies.**

I. INTRODUCTION

The engineering of systems [18] and the management of the process [19] to achieve that is addressed in the subject area of systems engineering [5], [4]. This makes the subject of systems engineering falling in the field of engineering as well as management sciences as noted in [17]. Currently, no theories are known in systems engineering that are based on the dual character of the subject, i.e. a strong mathematical foundation combined with management theory.

This paper proposes a theory to enable the development of mathematical models describing the systems engineering process. In future, this theory should be expanded to include the management of the process over an entire system life cycle to address the broader topic of the engineering of systems. In this paper, the term 'theory' is used in the tradition of philosophy of science as treated by the structuralist programme [6], [27], [2], [3], [15]. In the structuralist programme a theory is a mathematical structure containing sets, relations, and some axioms. Doeben-Henisch et.al. [13] made a first attempt to apply this theory concept in describing an engineering process.

This is a more formal form of treating research compared to the current research practice in systems engineering which includes the usual case studies [8], the proposed quasi-scientific methods with an experimental group and control group [26], or the usual empirical studies as performed in management science [28]. The current research practice describes strong or weak statistical correlations between parameters based on hypotheses derived from descriptions [8] and without formal mathematical models.

There are three possible aspects to a research project in engineering and technology management [8]:

1) *Testing* of *existing* theories, models and methods.
2) *Application* of existing theories, models and methods to a *new* problem.
3) *Construction* of *new* or improved theories, models and methods.

The current agenda for the development of the intended theory is first to achieve the testing of existing theories, models and methods in systems engineering through simulation that are validated with empirical measurements from systems engineering practice. These validated models can then be used to study the dynamic characteristics of the systems engineering process.

A. Process context

The scope of systems engineering activities in [9] consists of three main activities, i.e. systems engineering process, development phasing and system life cycle integration. This paper focuses on the development of a mathematical theory for the systems engineering process. The other two activities are not addressed here but the mathematical formalisms for development phasing and system life cycle integration should be addressed in future work. Further, the results of interactions between the activities (baselining, life cycle planning and integrated teaming [9]) should also be treated as mathematical formalisms in future. If all these identified mathematical formalisms would be combined a theory for Systems Engineering or Systems Engineering Management can be formulated in terms of the model presented in [9].

In a simple form, the systems engineering process consists of four subprocesses, i.e. requirements analysis, functional analysis and synthesis with the support of the systems analysis subprocess [9]. International standards [19], [18] and discipline handbooks [4], [5], [22] have augmented the four subprocesses with refinements to the main subprocesses and further description of implementation, verification, validation, support and management processes to achieve the engineering of systems.

II. PREVIOUS FORMALISM

A first formalism for an engineering process has been done in [13] as per Figure 1. The elements in this figure are characterized as follows [13]:

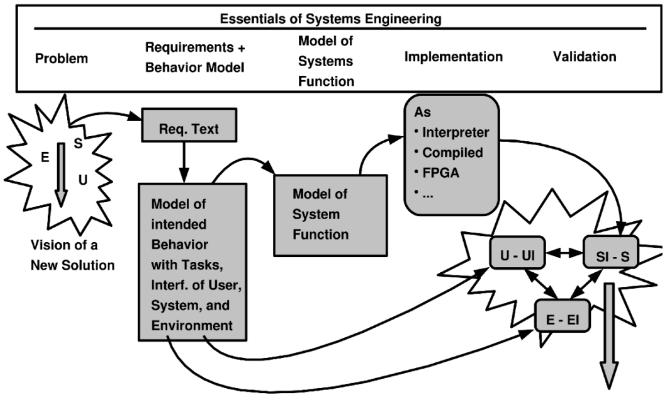

Fig. 1. Selected Elements of the Engineering Process [13]

"The process starts with a problem $\mathcal{P}$ *of a stakeholder. Through a communication process the systems engineer translates* $\mathcal{P}$ *into a* behaviour model $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$[1] *that represents the complete expected behaviour of the system to be designed:*

$$requirements \quad : \quad \mathcal{P} \longrightarrow \mathcal{M}_{\mathcal{S}-\mathcal{R}} \tag{1}$$

It is a known problem that this communication includes the semantic gap *problem which is rooted in the communication between the stakeholder and the engineer and the inherent difficulty to clarify the* meaning *of the terms used during the communication [12]. Based on* $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$*, the systems engineer develops during the design a* system model $\mathcal{M}_{\mathcal{SyS}}$ *which has to be verified:*

$$\begin{aligned} design \quad &: \quad \mathcal{M}_{\mathcal{S}-\mathcal{R}} \longrightarrow \mathcal{M}_{\mathcal{SyS}} && (2)\\ verification \quad &: \quad \mathcal{M}_{\mathcal{S}-\mathcal{R}} \times \mathcal{M}_{\mathcal{SyS}} \longrightarrow \mathcal{V} && (3)\end{aligned}$$

The $\mathcal{M}_{\mathcal{SyS}}$ *is converted into a* real system $\mathcal{M}_{\mathcal{SyS}*}$ *which again has to be validated.* Validation *is realized as a measurement process:*

$$\begin{aligned} implementation \quad &: \quad \mathcal{M}_{\mathcal{SyS}} \longrightarrow \mathcal{M}_{\mathcal{SyS}*} && (4)\\ validation \quad &: \quad \mathcal{P} \times \mathcal{M}_{\mathcal{S}-\mathcal{R}} \times \mathcal{M}_{\mathcal{SyS}*} \longmapsto \mathcal{V} && (5)\end{aligned}$$

where $\mathcal{V}$ *is a set of validation values indicating the correlation between the behaviour model* $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$ *and the system model* $\mathcal{M}_{\mathcal{SyS}}$*.*

The process to convert $\mathcal{P}$ *(in the* non-symbolic *space) into formalized requirements* $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$ *(in the* symbolic *space) and the* symbolic *system model* $\mathcal{M}_{\mathcal{SyS}}$ *into the* real *system* $\mathcal{M}_{\mathcal{SyS}*}$ *cannot be fully automated, because full automation is restricted to the symbolic space. The challenge of relating symbolic and non-symbolic spaces with each other also occurs during validation, when non-symbolic objects are compared with a symbolic description [12].*

The general structure of the behaviour model $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$ *can be described as a sequence of combined states* $\langle z_0, ..., z_f \rangle$*. A combined state* z *is defined by the driving task set* Γ*, the participating surfaces of the user called* user interface (UI)*, the intended* system interface (SI)*, and the assumed* environment interface (EI)*, thus,* $z_i \in Z \subseteq \Gamma \times INTF_U \times INTF_S \times INTF_E$*. A state change from a state* z_i *to a state* z_{i+1} *is caused by an action* $\alpha_i \in ACT \subseteq Z \times Z$*. Every sequence* p *of states for which it holds that* $(z_i, z_{i+1}) \in \alpha_i$ *is called a* usage process *or short* behaviour *of the behaviour model. The complete set of all possible behaviours of* $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$ *is described by the generating function* δ *that maps a start state* z_0 *into the possible usage processes ending in the* final states *or* goal states*. A complete* behaviour model $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$ *can then be defined as*

$$\mathcal{M}_{\mathcal{S}-\mathcal{R}} = \langle \Gamma, INTERF_{E/U/S}, Z, ACT, \delta, S, G_F \rangle \tag{6}$$

where $G_F \subseteq Z$ *is a set of goal states which shall be reached starting with the beginning state* S*.*

The constraints induced by the systems engineering process challenge the systems engineer to specify the required properties of a system in terms of its observable behaviour, including the interactions with the users and the environment."

A nontrivial aspect of this modeling is the interpretation of the task set Γ at least by the user U. This presupposes that a single task $\tau_i \in \Gamma$ is given as some string written in some language L_Γ which can be interpreted by the user U. Usually is this interpretation not part of the behaviour model. But, with regard to training and testing of users it could be necessary to include a complete specification of the language L_Γ as well as their intended interpretation $\mathcal{I}$ by the user. The semantic of the language L_Γ has as its domain of reference the complete behaviour model $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$.

III. CONSTRUCTING THE THEORY

The previous formalism for the description of the engineering process above is not sufficient. Terms used in it are still restricted to *primary objects* of the systems engineering process (i.e. problems, behaviour models, design models, etc.) but the main actors of this process, i.e. the stakeholders and discipline experts, are not included. To rectify this critical oversight, the formalism must be expanded. This expansion is treated in the following subsections.

A. The formalism

The main difference between the current proposed formalism and the previous formalism is the inclusion of the main actors of the systems engineering process, i.e. the *stakeholders* $\mathcal{S}$ and the different *discipline experts* $\mathcal{E}$. Additionally there can be different kinds of knowledge encoded in *documents*[2] $\mathcal{D}$ as well as *support systems* $\mathcal{A}$ used as enablers in the systems engineering process. The original requirements operator in (1) is extended to

---

[1] The $\mathcal{S}-\mathcal{R}$ index reminds one of the stimulus-response paradigm from the experimental behaviour sciences (cf. [25], [7]).

[2] Document is used in the widest sense of the word.

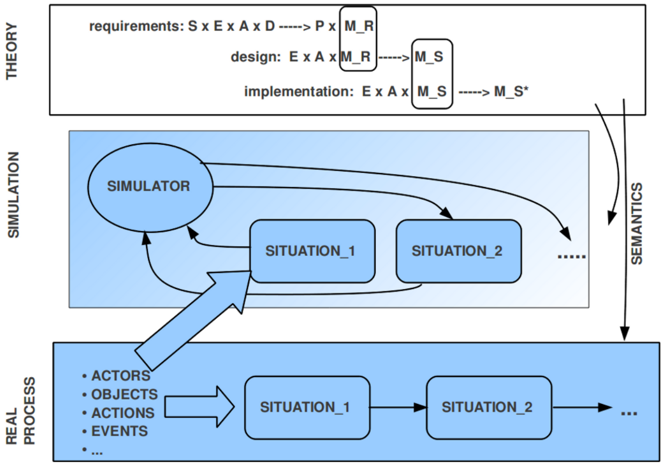

Fig. 2. Semantics of the formalism for the systems engineering process

$$requirements \quad : \quad \mathcal{S} \times \mathcal{E} \times \mathcal{A} \times \mathcal{D} \longrightarrow \mathcal{P} \times \mathcal{M}_{\mathcal{R}} \tag{7}$$

where the symbol $\mathcal{M}_{\mathcal{R}}$ means *functional* requirements $\mathcal{M}_{\mathcal{F}}$ and *non-functional* requirements $\mathcal{M}_{\mathcal{NF}}$ with the behaviour models $\mathcal{M}_{\mathcal{S}-\mathcal{R}}$ assumed to be equivalent to $\mathcal{M}_{\mathcal{F}}$, thus

$$\mathcal{M}_{\mathcal{R}} \quad = \quad \mathcal{M}_{\mathcal{F}} \cup \mathcal{M}_{\mathcal{NF}} \tag{8}$$

Thus, the problem description $\mathcal{P}$ together with $\mathcal{M}_{\mathcal{R}}$ are the *products* of the requirements analysis subprocess. The enablers for the process are the stakeholders, experts, support systems and additional documents. With a similar argument, the operators in (2) and (4) are expanded:

$$\begin{aligned} design \quad &: \quad \mathcal{E} \times \mathcal{A} \times \mathcal{M}_{\mathcal{R}} \longrightarrow \mathcal{M}_{\mathcal{S}} && (9) \\ implement \quad &: \quad \mathcal{E} \times \mathcal{A} \times \mathcal{M}_{\mathcal{S}} \times \mathcal{M}_{\mathcal{NF}} \longrightarrow \mathcal{M}_{\mathcal{S}*} && (10) \end{aligned}$$

Here $\mathcal{M}_{\mathcal{S}}$ replaces the previously used $\mathcal{M}_{\mathcal{SYS}}$. Based on these formalizations one can introduce the *theory for the systems engineering process* as

$$\Sigma(x) \text{ iff } \langle \mathcal{S}, \mathcal{E}, \mathcal{A}, \mathcal{D}, \mathcal{P}, \mathcal{M}_{\mathcal{R}}, \mathcal{M}_{\mathcal{S}}, \mathcal{M}_{\mathcal{S}*}, \rho, \delta, \iota \rangle \tag{11}$$

where the operators are abbreviated as follows:

$$\begin{aligned} \rho \quad &:= \quad requirements \\ \delta \quad &:= \quad design \\ \iota \quad &:= \quad implement \end{aligned}$$

But, for turning (11) into an *empirical* theory, one has to provide a mapping (called *semantics*) from these formulas into the intended domains. This will be achieved twofold:

- A mapping will be introduced incrementally from the formal parts of the theory into a real domain of those actors, objects, events, and actions which is commonly understood as the systems engineering process.
- Another mapping will be introduced incrementally into a defined simulation process, which in turn should be sufficiently similar[3] to a real-world systems engineering processes.

B. The semantics

The above formalism of the theory gets its meaning from the underlying empirical processes of real-world engineering. This includes a multitude of complex phenomena. Some of these phenomena include:

1) The main actors of the systems engineering process, i.e. *stakeholders* $\mathcal{S}$ and the *discipline experts* $\mathcal{E}$.
2) Besides the discipline experts $\mathcal{E}$ one has to assume an additional set of *support systems* $\mathcal{A}$.
3) There are different kinds of knowledge sources indicated in the theory as *documents* $\mathcal{D}$ used by the actors and support systems.
4) Usually there is not one but *multiple processes and subprocesses* active at a time.
5) The processes can be *distributed* over many physical or logical locations.
6) Processes can execute *synchronously* or *asynchronously*, *periodically* or *aperiodically*.
7) Processes can execute in *parallel* or *sequentially*.
8) Process can execute *iteratively*.
9) Different kinds of *communication* exists between processes.
10) Human communication processes are mostly mediated through *languages* which have open *meanings* embedded in open *grammars*.
11) The actions of humans are based on *internal mental models* which are the outcome of individual *learning processes*.
12) Human behaviour is influenced by internal *motivations/ emotions/ drives* as well as *physiological states*.

From this it can be concluded that the *human factors* of the stakeholders and discipline experts play a major part in the outcomes and performance of the systems engineering process. The interfaces between subprocess, the humans and support systems along with the embedded communication implied them are therefore fundamental to the systems engineering process.

IV. RELATIONSHIP WITH CURRENT PRACTICE

The *requirements* mapping described in (7) corresponds to the requirements loop described in [9], while the *design* mapping in (9) corresponds to the design loop in [9].

The requirements loop consists of the execution of the requirements analysis and part of the functional analysis subprocesses [9]. The design loop consists of the execution of part of the functional analysis and synthesis subprocesses [9].

---

[3] The definition of being *sufficiently similar* will be addressed in the future.

The $implementation$ mapping described in (10) has not sufficiently been described as part of the systems engineering process in the past. The result of the systems engineering process is a data pack [19], [9] that specifies the necessary products and processes to be implemented during the manufacturing and test process [19]. In [18], though, is the implementation subprocess described as part of the technical processes in the life cycle for the engineering of systems which should not be confused for the systems engineering process.

A. Requirements Loop

The requirements loop is described in [19] by the requirements analysis subprocess (P_{RA}) that implicitly executes the functional context analysis subprocess (P_{FC}) as well. If P_{RA} is assumed to consist of two processes (P_{RA}^{-} and P_{RA}) that precedes and follows P_{FC}, then

$$requirements \quad = \quad P_{RA}^{+} \circ P_{FC} \circ P_{RA}^{-} \tag{12}$$

The requirements analysis subprocess transforms the problem of the stakeholder $\mathcal{P}$ into functional requirements $\mathcal{M}_{\mathcal{F}}$, including functional performances, and non-functional requirements $\mathcal{M}_{\mathcal{NF}}$, including the life cycle quality factors [19].

B. Design loop

The design loop is described in [19] by the functional decomposition (P_{FD}) and synthesis subprocesses (P_S) and can be expressed as:

$$design \quad = \quad P_S \circ P_{FD} \tag{13}$$

During the functional decomposition subprocess P_{FD}, the functional behaviour models in $\mathcal{M}_{\mathcal{F}}$ are further decomposed into a functional architecture in $\mathcal{M}_{\mathcal{S}}$ to make it possible to define alternative subfunction arrangements and sequences, their functional interfaces and their performance requirements [19]. A systems analysis is performed in order to select a balanced set of subfunctions and to allocate performance requirements in $\mathcal{M}_{\mathcal{F}}$ to subfunctions to assure that the preferred functional architecture satisfies the system requirements [19].

During synthesis (P_S), the functional architecture in $\mathcal{M}_{\mathcal{S}}$ is translated into a design architecture in $\mathcal{M}_{\mathcal{S}}$ that provides an arrangement of system elements (hardware, software or people), their decomposition, internal and external interfaces, and design constraints [19]. A preferred design solution is selected from a set of alternatives based on the associated cost, schedule, performance and risk implications using the systems analysis subprocess for assessing design alternatives.

C. Example Human Factors Engineering

Figure 3 illustrates the formal structure of a behaviour model $\mathcal{M}_{\mathcal{F}}$ as part of the general theory Σ which describes the *intended outcome* of the functional requirements analysis. As described in the common human machine interaction (HMI) literature [10], [20], [21], [14], HMI is part of the requirements and design processes for exploring the optimal parameters of the intended system interface with regard to the intended users and their tasks (including environmental factors). While this paradigm is well known and discussed in numerous publications it should be realized that within the intended systems engineering theory Σ this HMI-view should also be applied on the acting engineers themselves. In the primary view of the Σ-theory the systems engineering experts $\mathcal{E}$, the stakeholders $\mathcal{S}$, and even the supporting systems $\mathcal{A}$ are the 'users' which interact with each other in an engineering environment. Their task is to develop a new system and the systems engineering process should be supported in this. This includes that the *engineers as the main actors* need an appropriate mental model which enables them to find the adequate solution. The above mentioned *simulation* dimension (see also Figure 1) can be viewed from the perspective of the acting experts and stakeholders. The simulations have to support these actors and should be *smart* and *adaptive* to be able to give advice when needed. Future expanded version of the Σ-theory will include an additional 'user interface' to the whole process for the management aspects of the systems engineering process.

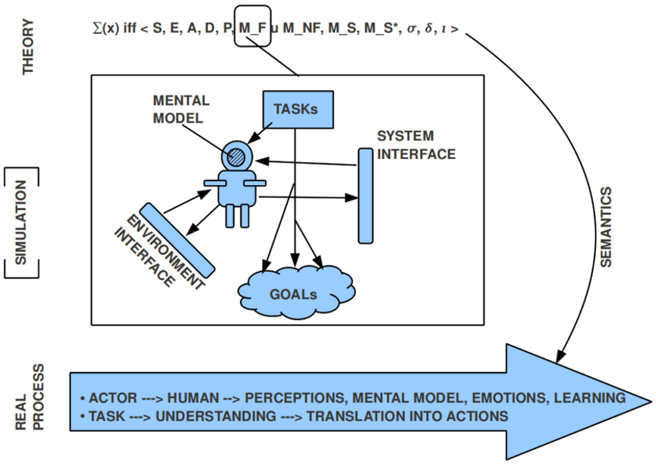

Fig. 3. Human-Factors Engineering Example

V. CONCLUSION

In this paper a programme for the development of a structuralist theory is presented for the engineering of systems. An important aspect in the formulation of this theory is the inclusion of people as part of the systems engineering process. Although assumptions are made to simplify the initial formulation of the theory, it can be stated that the engineering of systems is a complex topic that needs this theory.

Further work still needs to be done to bring the theory closer to current practice. A research programme will be formulated based on this theory and the programme status will be reported on the website http://www.os-pe.org. Empirical research into practices used in industry will also be used to calibrate the theory. Further development of the theory will also be undertaken to address the identified short comings and simplifying assumption pointed out in this paper.

Some of the models can be implemented with stock and flow models as used in the systems dynamics arena [23].

In general the models can be implemented on the OKSIMO simulator platform [11], [24] as the formulation of the theory in this paper is compatible with the underlying theory of the OKSIMO simulation platform. It is the intension also to cooperate with the development team of OKSIMO to make simulations of the theory in this paper possible.

This theory for the engineering of systems is a contribution for a better understanding how to establish a theoretical basis for systems engineering and engineering management.

REFERENCES

[1] Alexander, I.F. & Stevens, R.; Writing Better Requirements. Addison-Wiley, 2002.

[2] Balzer, W., *Empirische Theorien: Modelle, Strukturen, Beispiele*, Wiesbaden (Germany): Friedr.Vieweg & Sohn, 1982

[3] Balzer, W.; Moulines, C. U.; Sneed, J. D., *An Architectonic for Science*, Dordrecht (NL): D.Reidel Publishing Company, 1987

[4] Blanchard B.S., System Engineering Management, Wiley, 4th edition, 2008.

[5] Blanchard B.S. & Fabrycky W.J., Systems Engineering and Analysis, Prentice-Hall, 5th edition, 2010.

[6] Bourbaki, N., *Theorie Des Ensembles*, Paris: Hermann, 1970

[7] Bower, G.H; Hilgard E.R. *Theories of learning*, 5th. ed., Prentice-Hall Inc., 1981 (with a german edition: *Theorien des Lernens, Vol.1+2*, 5th. rev.ed., Stuttgart (Germany): Klett-Cotta, 1983, translated by Aebli, H.; Aeschbache , 1983, 1984)

[8] Bus, A.; Research Guide For Post-Graduate Students. Issue 18. Graduate School for Technology Management, University of Pretoria, South Africa, 2007.

[9] Defense Acquisition University; Systems Engineering Fundamentals. Defense Acquisition University Press, Fort Belvoir V.A., USA, 2001.

[10] Dix, A.; Finlay,J.E.; Abowd,G.D.; Beale, R.: *Human-Computer Interaction*. Pearson Education, 2003, 3rd.ed.

[11] Doeben-Henisch, G.; *The planet Earth simulator project - a case study in computational semiotics*, AFRICON, 2004. 7th AFRICON Conference in Africa, Publication Year: 2004 , Page(s): 417 - 422 Vol.1

[12] Doeben-Henisch, G.; Wagner, M. *Validation within Safety Critical Systems Engineering from a Computational Semiotics Point of View*, In: IEEE Africon2007 Intern.Conference, Windhoek (Namibia), Sept.2007

[13] Doeben-Henisch, G.; Bauer-Wersing, U.; Erasmus, L.; Schrader,U.; Wagner, W. *Interdisciplinary Engineering of Intelligent Systems. Some Methodological Issues*, part of the 'Proceedings of the International Workshop Modelling Adaptive And Cognitive Systems (ADAPCOG 2008)' , Joint Conferences of SBIA'2008 (the 19th Brazilian Symposium on Artificial Intelligence); Salvador (Brazil) Oct-26 - Oct-30, 2008. As chapter in the E-Book: Angelo Loula & João Queiroz (Eds.), *ADVANCES IN MODELING ADAPTIVE AND COGNITIVE SYSTEMS*, URL: http://www2.uefs.br/graco/amacs/

[14] Dumas, J.S.; J.E.Fox, *Usability Testing: Current Practice and Future Directions*, In: Sears, A.; Jacko, J.A. (Eds.): *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies and Emerging Applications.*, New York - London: Lawrence Erlbaum Associates, Publ., 2008, 2nd.ed., pp.1129 - 1150.

[15] Erasmus, L.D. *The Formulation and Evaluation of a Neuron Model Based on Biological Neurons*, PhD-Thesis, Potchefstroom: North West University, 2007

[16] Graduate School of Technology Management. Online: http://www.up.ac.za/gstm (Accessed on 24 March 2011).

[17] INCOSE, Systems Engineering Hanbook - A "How to" guide for all engineers. Version 2.0 International Council on Systems Engineering, July 2007.

[18] ISO/IEC 15288:2007, System Life Cycle Processes. International Organisation for Standards, Geneva, Switzerland, 2007.

[19] ISO 26702:2007/IEEE Std 1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process. International Organisation for Standards, Geneva, Switzerland, 2007.

[20] Lauesen, S.: *Task Descriptions as Functional Requirements*, in: IEEE Software 2003, March/April, pp. 58-65.

[21] Lauesen, S.: *User Interface Design. A software Engineering Perspective.* London et al:Pearson - Addison Wesley, 2005. See also the Webpage with additional Material: http://www.itu.dk/ slauesen/SorenUID.html

[22] Martin, J.N., Systems Engineering Guidebook, A Process for Developing Systems and Products. CRC Press, Boca Raton, USA, 1997.

[23] Morecroft, J. Strategic Modelling and Business Dynamics: A Feedback Systems Approach. John Wiley & Sons, 2007.

[24] Open Knowledge Simulation Modeling. Online: http://oksimo.inm.de/ (Accessed on 24 March 2011).

[25] Skinner, B.F., *Science and Human Behavior*, New York et al: The Free Press, 1953

[26] Sparrius, A. Is system engineering a cargo-cult science? INCOSE SA Conference Proceedings, Pretoria, South Africa, August 2010.

[27] Suppe, F. (Ed.), *The Structure of Scientific Theories*, 2nd. ed., Urbana: University of Illinois Press, 1979

[28] Valerdi, R. and Davidz, H. L. (2009), Empirical research in systems engineering: challenges and opportunities of a new frontier. Systems Engineering, 12: 169181. doi: 10.1002/sys.20117