

Integration and Framing between System Engineering, Enterprise Engineering and Whole of Society

Louwrence Erasmus
Postgraduate School of Engineering
Management, University of Johannesburg and
ISG, DPSS, CSIR, Pretoria, South Africa
+27 (83) 540 5092
l.erasmus@ieee.org

Braam Greeff
Integrative Systems Group, DPSS, Council for
Scientific and Industrial Research, Pretoria
+27 (82) 904 4352
BGreeff@csir.co.za

Copyright © 2017 by Dr LD Erasmus and AP Greeff. Published and used by INCOSE with permission.

Abstract. This paper proposes the integration between the domains of product development, systems engineering, enterprise engineering & architecture and the whole of society. The effort addresses the following systems engineering imperatives as per INCOSE Vision 2025: “Expanding the application of systems engineering across industry domains, applying systems engineering to help shape policy related to social and natural systems, expanding the theoretical foundation for systems engineering, advancing the tools and methods to address complexity.” Formal theoretical structures in the form of mathematical predicates are used from the structuralist programme in the philosophy of science to frame the integration. The formal structures make future simulations of the identified integration possible.

Introduction

This paper describes how the Integrative Systems Group (ISG) of the Defense, Peace, Safety and Security (DPSS) business unit and the Council for Scientific and Industrial Research (CSIR) in Pretoria, South Africa proposes the integration of the following systems engineering imperatives of the INCOSE Vision 2025:

- Expanding the application of systems engineering across industry domains.
- Applying systems engineering to help shape policy related to social and natural systems.
- Expanding the theoretical foundation for systems engineering.
- Advancing the tools and methods to address complexity.

The view in this paper is that of a systems engineer. Systems thinking is applied to address the objectives above and using the concepts from the structuralist programme in the philosophy of science (Bourbaki, 1970), (Suppe, 1979), (Balzer, 1982), (Balzer, et al., 1987), (Erasmus, 2007).

Systems Hierarchy Levels

Traditionally, systems engineering is applied in military development projects in South Africa. Systems hierarchy levels are defined for the systems acquisition process used in the South African National Defence Force (De Waal & Buys, 2007). The systems hierarchy levels are summarised as shown in Figure 1 using a telecommunications example.

Type	Level Summary	Systems Hierarchy Level	Description	Type of Capability	Example
Virtual Systems	National and International Levels	Level 9-10	Multi-govern ment and societal systems	Self-Organising Operational Capability	International Telecommunications Union (ITU), the interministerial committee on information and communication technologies (ICT), etc.
		Level 8	Joint Higher Order Organisations	Coordinated Operational Capability	Telecommunications Corporation, National Department of Telecommunications
	Service Delivery, Backstage and Support	Level 7	Operational System	Operational Capability	Cellular Telecommunications Business Division or Company
Physical Systems	Service Delivery, Backstage and Support	Level 6	Core System	Core Capability	Video Streaming Service Delivery with personnel, organizational structure, support systems, training of personnel, equipment, business processes, operational facilities, operational business information, technology roadmap, budget, business model.
		Level 5	Products System	Pseudo Capability	LTE-A Equipment System, Equipment Repair Workshops, Servicing Facilities, Training Materials and Facilities, etc.
	Technology, Support Facilities and Equipment	Level 2-4	Products		Smartphones consisting of software, CPU, RF module, power supply, etc.
		Level 1	Raw Material		Aluminium, Plastics, Ruby crystal glass, etc.

Source: Adapted from Van der Walt & Erasmus (2016)

Figure 1. Systems Hierarchy

De Waal & Buys (2007) has shown that these levels have a correspondence with the semantic theoretical constructs of systems levels of Boulding's General Systems Theory. The construct of systems hierarchy levels addresses the progression from complicated engineered levels to

the complexity of human interaction with engineered technological systems and the organisation of society and subsets of society around these technologies. From a discussion with the researchers, it should be noted that each systems hierarchy level requires a different epistemology for scientific study.

Policies relating to social and natural systems are formulated for impacting directly the social organisation around technologies on systems hierarchy levels 7 to 10 in Figure 1.

Systems engineering has also been applied in the nuclear energy industry (Theron, et al., 2007), the mining sector (Van Zyl & Lotz, 2004) and healthcare (Van der Walt & Erasmus, 2016). Lately, systems engineering is applied in a variety of sectors by just looking at reports on its application at the INCOSE South Africa Chapter's annual conference at http://incose.org.za/Conference_Proceedings.

Formal Structures

The theoretical foundation of systems engineering was expanded through the use formal mathematical structures and logical predicates in:

- A theory of the systems engineering process (Doeben-Henisch, et al., 2008) (Erasmus & Doeben-Henisch, 2011a)
- A theory of systems engineering management (SEMBASE) (Erasmus & Doeben-Henisch, 2011b).

In the structuralist programme a theory T is a mathematical structure containing sets of objects O , relations between objects R and axioms A defined as a structure using the following predicate logic form (Erasmus, 2007):

$$T(x) \Leftrightarrow x \in \langle O, R, A \rangle$$

where

- $O = \{o_1, \dots, o_n\}$ represents the set of objects
- $R = \{r_1, \dots, r_m, f_1, \dots, f_k\}$ represents the set of relationships r_i , mappings and functions f_i
- $A = \{a_1, \dots, a_l\}$ represents axioms a_i of the theory T

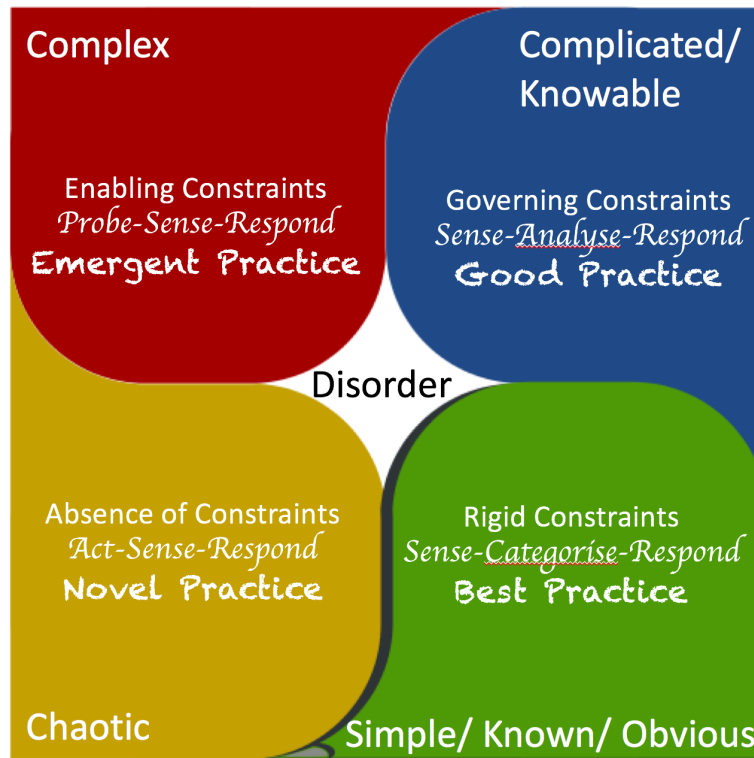
The description of the theory structure is similar to that for an ontology (Sarder & Ferreira, 2007). The theory structures can provide a sound basis for empirical ontologies as discussed in (Honour & Valderi, 2006) (Sarder & Ferreira, 2007), (Graves & West, 2012) and (Orellana & Madni, 2014).

Further, a formal theory structure can be implemented in software using the predicate logic directly in Prolog or transforming the predicate logic clauses into systems equations (Erasmus, 2007).

Cynefin Framework

In dealing with the introduced complexities on systems hierarchy level 6 and higher in Figure 1, a way to make sense from the complex constructs is necessary. Also, the combination of the theory structure with sense making in complexity makes it possible to simulate the dynamics in the complex domain using a platform like OKSIMO (<http://www.inm.de/index.cfm?siteid=32>), or for more simpler models classic system dynamics can be employed.

The Cynefin Framework is such a way, amongst others, to make sense of the world around us (Snowden, 2002) (Snowden & Boone, 2007). Figure 2 shows a redrawn Cynefin Framework.



Source: Redrawn from (Snowden & Boone, 2007)

Figure 2. Cynefin Framework

In summary, there exist three types of systems, namely chaotic, complex and ordered. The ordered domain is further subdivided into complicated and simple systems. The Cynefin Framework consists out of five domains, namely unordered, chaotic, complex, complicated and simple domains (Snowden, 2002).

Simple domain. Also known as the domain of knowable or obvious knowledge. The conduct for this domain is first to sense the signals from the system as data, then categorise the sensed data and respond accordingly by best practice and adhering to rigid constraints (Snowden & Boone, 2007). Formally this can be expressed as

$$T_S(x) \Leftrightarrow x \in \langle O_S, R_S, A_S \rangle$$

where

- $O_S = \{\vec{o}_1, \dots, \vec{o}_p, o_1, \dots, o_n\}$ represents the set of objects of the system and its environment, specifically \vec{o}_i is the commanding elements of the system and o_j the interconnected elements other than the commanding elements.
- $R_S = \{r_1, \dots, r_m, f_1, \dots, f_k\}$, $r_i \in \{x | o_j \sigma_{jk} o_k, \forall j, k \text{ and } \forall \sigma_{jk} \text{ is a relation and } \sigma_{jk} \neq \emptyset \text{ and } o_j, o_k \in \vec{O}_K\}$, $f_i \in \{Best Practice\}$, $\vec{O}_K \subset O_S$ is the set of commanding elements.
- $A_K = \{a_1, \dots, a_l\}$ $a_i \in \{Rigid constraints\}$

Complicated domain. Also known as the knowable domain. The conduct for this domain is first to sense the signals from the system as data, then analyse the sensed data and respond accordingly having governing constraints to guide the use of good practice (Snowden & Boone, 2007). Formally this can be expressed as

$$T_K(x) \Leftrightarrow x \in \langle O_K, R_K, A_K \rangle$$

where

- $O_K = \{\vec{o}_1, \dots, \vec{o}_p, o_1, \dots, o_n\}$ represents the set of objects of the system and its environment, specifically \vec{o}_i is the influencing/lever elements of the system and o_j the interconnected elements other than the influencing elements.
- $R_K = \{r_1, \dots, r_m, f_1, \dots, f_k\}$, $r_i \in \{x | o_j \sigma_{jk} o_k, \forall j, k \text{ and } \forall \sigma_{jk} \text{ is a relation and } \sigma_{jk} \neq \emptyset \text{ and } o_j, o_k \in O_K\}$, $f_i \in \{\text{Good Practice}\}$
- $A_K = \{a_1, \dots, a_l\}$ $a_i \in \{\text{Governing constraints}\}$

Complex domain. The conduct for this domain is first to probe the system and then sense the signals from the system and its environment as data, then respond accordingly having enabling constraints in using emergent practice (Snowden & Boone, 2007). Formally this can be expressed as

$$T_C(x) \Leftrightarrow x \in \langle O_C, R_C, A_C \rangle$$

where

- $O_C = \{o_1, \dots, o_n\}$ represents the set of objects of the system and its environment
- $R_C = \{r_1, \dots, r_m, f_1, \dots, f_k\}$, $r_i \in \{x | o_j \sigma_{jk} o_k, \forall j, k \text{ and } \forall \sigma_{jk} \text{ is a relation and } \sigma_{jk} \neq \emptyset \text{ and } o_j, o_k \in O_C\}$, $f_i \in \{\text{Emergent Practice}\}$
- $A_C = \{a_1, \dots, a_l\}$ $a_i \in \{\text{Enabling constraints}\}$

Chaotic domain. The conduct for this domain is first to act decisively then sense the signals from the system and its environment as data, then respond accordingly without constraints in using novel practice (Snowden & Boone, 2007). Formally this can be expressed as

$$T_{CX}(x) \Leftrightarrow x \in \langle O_{CX}, R_{CX}, A_{CX} \rangle$$

where

- $O_{CX} = \{o_1, \dots, o_n\}$ represents the set of objects of the system and its environment
- $R_{CX} = \{r_1, \dots, r_m, f_1, \dots, f_k\}$ $r_i \in \emptyset$, $f_i \in \{\text{Novel Practice}\}$
- $A_{CX} = \emptyset$

Disorder. In this domain “ignorance is bliss”, and one is not aware of anything (Snowden & Boone, 2007). Formally this can be expressed as

$$T_D(x) \Leftrightarrow x \in \emptyset$$

In the real world, one can be in all five domains simultaneously. This is a pluralistic understanding of the world. Thus, one’s understanding of the world can be expressed as the vector

$$T_{wu}(x) = \begin{bmatrix} T_S(x) \\ T_K(x) \\ T_C(x) \\ T_{CX}(x) \\ T_D(x) \end{bmatrix}$$

The above formalism can also be interpreted, in a similar way how Erasmus (2007) did it by transforming the formal theory structures into mathematical system representations used in control systems engineering. Just be aware that the states in the systems equations can be from any general type of mathematical space that includes numbers, characters, strings, and

other symbolic representations of physical objects (Erasmus, 2007). A summary for this transformation is illustrated Figure 3.

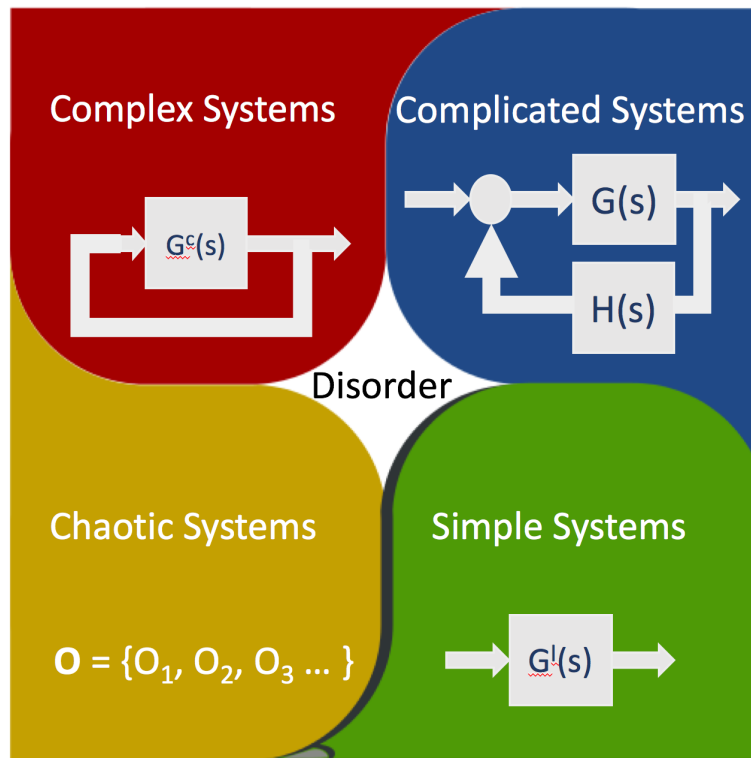
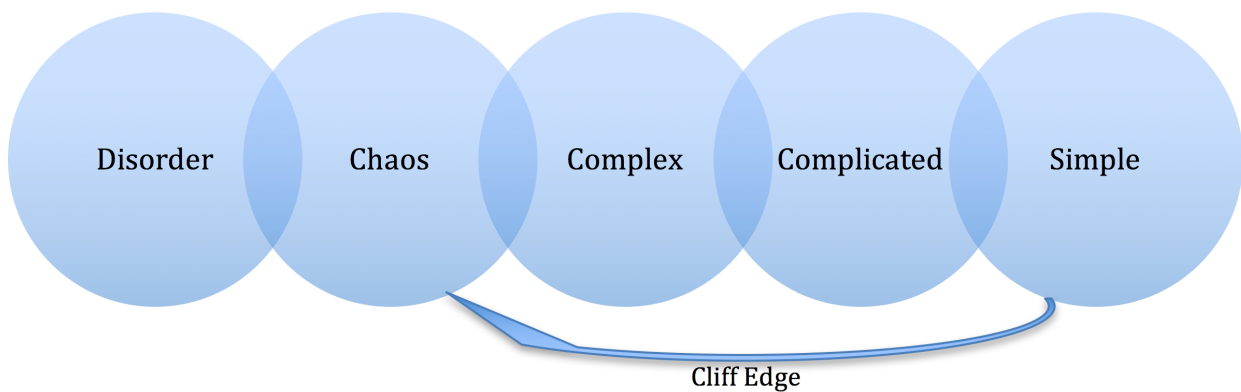


Figure 3. Cynefin Framework: Control Engineering View

For the further development of the argument in this paper, another view of the Cynefin Framework is useful as presented in Figure 4. Take note of the loopback from Simple to Chaos domains. The loopback is represented in Figures 2 and 3 as a cliff edge between the two domains.



Based on: Snowden & Boone (2007)

Figure 4. Cynefin Framework: Linear View

Integrating Product Development, System Engineering, Enterprise Engineering/Architecture and Whole of Society

Within ISG DPSS the following four domains of holistic development have been identified:

- Whole of Society Awareness.
- Enterprise Engineering & Enterprise Architecture.
- System Engineering/Engineering of a system (as opposed to the subject of systems engineering).

- Product Development.

These four areas overlap with interfaces as presented in Figure 5.

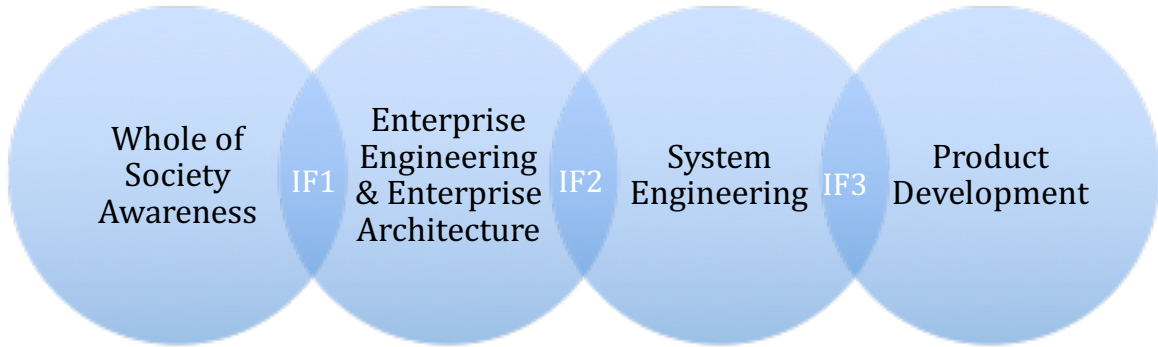


Figure 5. Interface Identification Diagram

The above four domains of holistic development are proposed to be roughly mapped to different parts of the Systems Hierarchy in Figure 1 in the following way:

- Whole of Society Awareness maps to levels 8, 9 and 10.
- Enterprise Engineering & Analysis maps to levels 6 to 8.
- System Engineering maps to levels 4 to 6
- Product Development maps to levels 2 to 4

Whole of Society Awareness is using techniques that are typically taking someone or group from a disordered domain into a chaotic domain and as soon as possible into the complexity domain. This means that to transform formal structures as follow:

$$T_D(x) \rightarrow T_{CX}(x) \rightarrow T_C(x)$$

The outcome of the actions of Whole of Society Awareness is opportunities and challenges x expressed in the complexity domain $T_C(x)$ for multi-government and societal systems as well as joint higher order organisations. The interface IF1 is the challenges and opportunities that consists out of those objects O_{IF1} and relationships R_{IF1} , where $\langle O_{IF1}, R_{IF1} \rangle \subseteq T_C(x) \cap T_K(x)$.

Enterprise Engineering is using techniques that are typically taking someone or a group from a complexity domain into a complicated domain:

$$T_C(x) \rightarrow T_K(x)$$

Enterprise Architecture is taking the understanding of a group from a complicated domain into the simplicity domain. This means that to transform formal structures as follow:

$$T_K(x) \rightarrow T_S(x)$$

The outcome of the actions of Enterprise Engineering & Analysis is required operational capabilities x expressed in the complexity domain $T_C(x)$ for core systems. The interface IF2 is the required operational capabilities that consists out of those objects O_{IF2} and relationships R_{IF2} , where $\langle O_{IF2}, R_{IF2} \rangle \subset T_C(x)$.

System Engineering is using techniques that are typically taking someone or group from a complexity domain into a complicated domain and sometimes into the simplicity domain. This means that to transform formal structures as follow:

$$T_C(x) \rightarrow T_K(x) \rightarrow T_S(x)$$

The outcome of the actions of System Engineering is product development specifications x expressed in the complicated domain $T_K(x)$ for products. The interface IF3 is the challenges and opportunities that consists out of those objects O_{IF3} and relationships R_{IF3} , where $\langle O_{IF3}, R_{IF3} \rangle \subseteq T_K(x)$.

Part of the discussion above can be summarised in Figure 6.

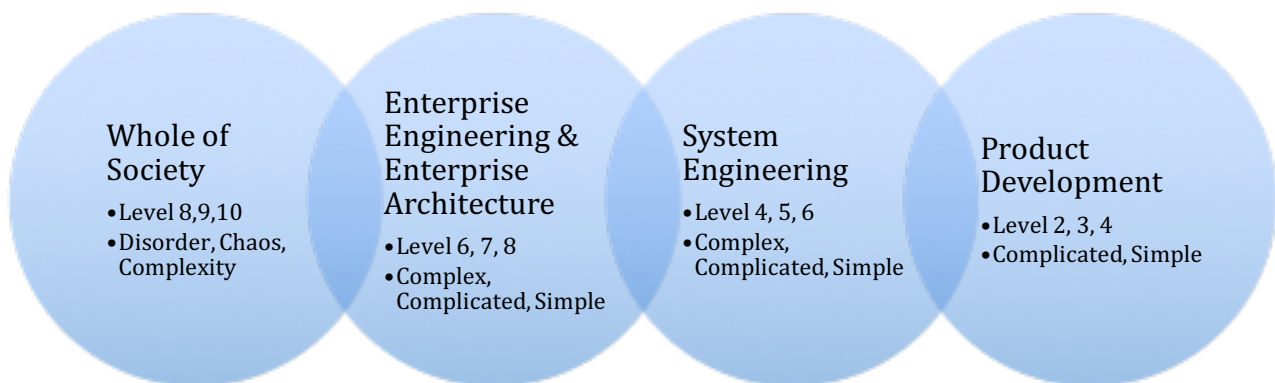


Figure 6: Relationship with Systems Hierarchy Levels and the Cynefin Domains

Integrating System Engineering, Enterprise Engineering/Architecture and Whole of Society into a System Life Cycle

This paper proposes that the DPSS holistic development approaches should be based on the adapted Vee-Model for Systems Life Cycle development (The Standard Model or TSM) shown in figure 7 and supported by putting all the above theory into practice.

The introduction of Phase -2 is to do all the necessary Whole of Society work, known in the government sector as Whole of Government. The deliverables include opportunities and challenges that can be used by Phase -1 that perform the Enterprise Architecture and Engineering.

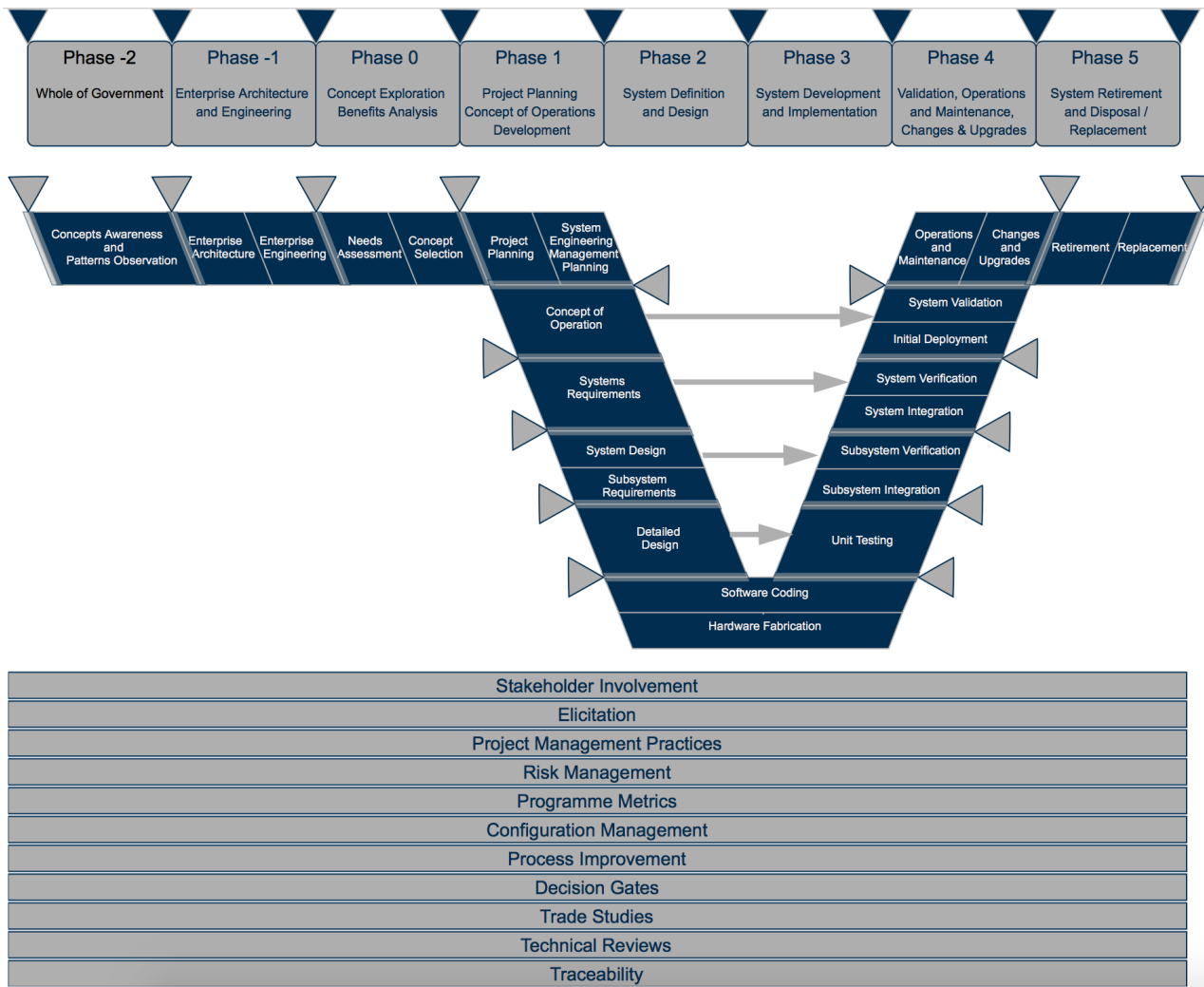
The input from Phase -1 includes a required operational capability that is co-evolved between Enterprise Engineers and Systems Engineers for input to Phase 0 for starting of the System Engineering Development effort.

Co-evolution between System Engineering and Product Development develops the Product Development Specification, with Systems Engineering keeping the responsibility.

The concept portal for deployment on the CSIR IntraWeb is finished. The concept needs to be implemented and populated by content. The content for systems engineering is based on the INCOSE Systems Engineering Handbook and enhanced with the experience of CSIR systems engineers.

The content for enterprise engineering and architecture is based on Layer Enterprise Architecture Development (LEAD) with a strong TOGAF influence for the architecting part.

The other domain areas have their content and can be integrated with this portal.



Based on: FHWA CA Division (2016)

Figure 7: CSIR's Vee-Model for a System Life Cycle

Further development of the theory structures and the accompanying semantics is necessary to eventually arrive at a platform to simulate the adapted Vee-Model for DPSS. The vision is a cyber-information-physical simulation of development processes to forecast the development outcomes for a specific tailoring of TSM. The vision is like the current German Industrie 4.0 and other advanced manufacturing programmes in the world for simulating cyber-physical manufacturing designs first before deploying them for implementation in reality.

Conclusion

By using a theoretical structure from the structuralist programme in philosophy, it was possible to derive in a consistent way a definition of what the interfaces are between the domains of Product Development, System Engineering, Enterprise Engineering & Architecture and Whole of Society Awareness.

This paper contributes to expanding the application of systems engineering across industry domains by proposing the establishment of a cyber-information-physical capability to evaluate virtually the outcomes of a tailored holistic development process spanning several domains. Further, the paper identifies the system hierarchy levels 7 and higher where the

application of systems engineering may help shape policy related to social and natural systems.

The further application of results from the Structuralist Programme in the philosophy of science to systems engineering problems helps to expand the theoretical foundation for systems engineering. A cyber-information-physical tool based on the theory structures is proposed to address complexity.

References

- Bourbaki, N., 1970. *Theorie Des Ensembles*. Paris: Hermann.
- Suppe, F., ed., 1979. *The Structure of Scientific Theories*. 2nd Edition ed. Urbana(III): University of Illinois Press.
- Balzer, W., 1982. *Empirische Theorien: Modelle, Strukturen, Beispiele*. Wiesbaden, Germany: Friedr. Vieweg & Sohn.
- Balzer, W., Moulines, C. & Sneed, J. D., 1987. *An Architectonic for Science*. Dordrecht: D.Reidel Publishing Company.
- Erasmus, L. D., 2007. *The Formulation and Evaluation of a Neuron Model Based on Biological Neurons*. PhD-Thesis ed. Potchefstroom: North-West University.
- Doeben-Henisch, G. et al., 2008. Interdisciplinary Engineering of Intelligent Systems. Some Methodological Issues. In: A. Loula & J. Queiroz, eds. *ADVANCES IN MODELING ADAPTIVE AND COGNITIVE SYSTEMS*. Salvador: e-book.
- Erasmus, L. D. & Doeben-Henisch, G., 2011a. *A Theory of the System Engineering Process*. Victoria, Zambia, IEEE.
- Erasmus, L. D. & Doeben-Henisch, G., 2011b. *A Theory for System Engineering Management*. Stellenbosch, South Africa, ISEM.
- Sarder, M. B. & Ferreira, S., 2007. *Developing Systems Engineering Ontologies*. San Antonio, TX, IEEE, pp. 1-6.
- Honour, E. C. & Valderi, R., 2006. *Advancing an Ontology for Systems Engineering to Allow Consistent Measurement*. Los Angeles, CA, INCOSE SERC, pp. 1-12.
- Graves, H. & West, M., 2012. *Current State of ontology in engineering systems*, Internet: OMG.
- Orellana, D. W. & Madni, A. M., 2014. Human System Integration Ontology: Enhancing Model Based Systems Engineering to Evaluate Human-system Performance. *Procedia Computer Science*, Volume 28, pp. 19-25.
- Theron, W., Matzner, H. & Erasmus, L., 2007. *The Pebble Bed Modular Reactor and the usage of systems engineering to establish new standards for the nuclear revival*. Pretoria, IFAC, pp. 1-7.
- Snowden, D., 2002. Complex acts of knowing: paradox and descriptive self-awareness. *Journal of Knowledge Management*, 6(2), pp. 100-111.
- Snowden, D. & Boone, M. E., 2007. A Leader's Framework for Decision Making. *Harvard Business Review*.
- Van der Walt, R. & Erasmus, L., 2016. *Healthcare Facility Commissioning – The Transition of Clinical Services*. Edinburg, Scotland, UK, INCOSE.
- De Waal, J. & Buys, A., 2007. Interoperability and Standardisation in the Department of Defence: An Exploratory Study. *Journal of Industrial Engineering*, Volume 18, pp. 175-190.
- Van Zyl, D. & Lotz, S., 2004. *Case study: Requirements engineering in the mining industry*. Pretoria, INCOSE SA.
- FHWA CA Division, 2016. *Systems Engineering Guidebook for ITS*. [Online] [Accessed 11 2016].

Biography

Louwrence Erasmus worked for more than 20 years on multi-disciplinary projects in academia, South African and international industries. He is a senior lecturer in the Postgraduate School of Engineering Management at the University of Johannesburg. He is a Principal Systems Engineer at the CSIR since 2013. He is a research associate at the Graduate School of Technology Management at the University of Pretoria. He is an advisory board member of Third Circle Asset Management. He graduated from the Potchefstroom University with the B.Sc., B.Eng., and M.Sc. degrees in 1989, 1991, 1993 and was awarded a PhD degree in 2008 from North West University, Potchefstroom. He is a registered professional engineer with ECSA and a senior member of IEEE and SAIEE. His interest is formal structures using constructivist philosophy of science and their practical implications in the practice of systems engineering.

Braam Greeff is the Research Group Leader for Systems Engineering at CSIR's Integrative Systems Group. He has more than 25 years experience in the development of various software-, electronic- and mechanical product systems within the Defense and Aerospace domain. He holds a B.Eng. (Electronics), M.Eng. (Engineering Management) from the University of Pretoria. He was a president of INCOSE South African Chapter.