TOWARDS UNIFICATION OF PRODUCT AND ENTERPRISE SYSTEM DESCRIPTIONS

JONNRO ERASMUS
Council for Scientific and Industrial Research, Integrative Systems Group, South Africa
jerasmus1@csir.co.za (Corresponding)

LOUWRENCE D ERASMUS
Council for Scientific and Industrial Research, Integrative Systems Group, South Africa
lerasmus1@csir.co.za

JAN-HARM C PRETORIUS
University of Johannesburg, Postgraduate School of Engineering Management, South Africa
jhcpretorius@uj.ac.za

ABSTRACT

An argument is presented for the unification of descriptions of product systems and enterprise systems. Product systems are developed and produced by enterprises, thus forming an integral part of the enterprise's architecture. However, many products are utilised by enterprises and some product systems contain entire businesses, such as the operating and maintenance business of a power station. Thus, products are part of enterprises, but enterprises may also be part of product systems. To enable the design of systems that include the product, its user and all the enterprises that make the product available and possible, it is necessary to align the enterprise engineering and systems engineering views. This article presents a starting point that allows the two disciplines to more accurately refer to a specific element of the complete system-of-interest. The aim is that this will allow for improved communication between the practitioners of the different disciplines and perhaps the development of improved solutions.

Key words: Systems hierarchy, enterprise architecture, ontology

INTRODUCTION

One of the fundamental goals of systems engineering is to develop products and services that not only satisfy requirements, but rather fulfil expectations (INCOSE, 2011). It is for this reason that so much attention and emphasis is placed on needs and requirements analysis, to ensure that it is truly understood and agreed what is expected of the system, how it will be used, supported and eventually retired (International Organization for Standardization et al., 2011). It is well understood that the product or service cannot be designed in isolation, but that its context, environment and user should be considered as part of the solution development (Sage and Cuppan, 2001; Erasmus and Doeben-Henisch, 2011; Katina et al., 2014). Great progress has been made in the consideration of human factors in systems, but the more holistic approach, i.e. the enterprise view, is still considered a separate endeavour. Yet, many products are not utilised and supported by individuals, but rather by complete businesses, with their own competencies, practices, techniques, tools and infrastructure. A power station is clearly not only the hardware that generates the electricity and software that protects the hardware. A large power station can be quite labour intensive and will inevitably feature all the usual business activities, such as supply chain management, enterprise resource planning and customer relationship management.

Selected international standards and good practice guides, such as IEEE Std 1220-2005, do mention the need to consider the enabling lifecycle processes as part of the engineering effort (IEEE Computer Society et al., 2005). However, it provides no further information on how the system-of-interest and its enabling processes may be integrated to deliver a system of systems that satisfies all input requirements. Thus, a need exists for an approach that includes the design of the product system and the business(es) that will utilise and support the system. To allow for effective communication and collaboration between the systems and enterprise engineering disciplines, it is necessary to create a reference framework. Such a framework should be discreet enough to enable clear understanding of the system level or enterprise area currently under consideration.

The purpose of this article is not to unify the domains of systems and enterprise engineering, but rather to create a framework that will allow for integrated design of the product and business(es) that the system comprise of. To create such a common framework, a few widely used ways to describe product systems and enterprise systems will be explored, to identify areas of overlap and points of commonality. A simple framework is then presented and a few conclusions are reached.

APPLICABILITY

A system changes as it progresses through its lifecycle stages. A product is generally only a collection of knowledge and information during the design and development stage, transformed into a physical entity during manufacturing or implementation. Additionally, system elements are typically treated differently based on its perceived position in the system hierarchy. These themes are explored briefly to allow for precise reference to the types of systems considered in this article.

System lifecycle

The lifecycle phase of the system-of-interest is an important consideration, because some of its attributes change as it progresses through its lifecycle. For example, control of the system changes ownership as it advances through its lifecycle. During the development phase, engineering is in control of the product, who then hands it over to manufacturing or construction. Once the system is delivered to customers, it may be utilised without control of the original owner (Stark, 2011). A system also generally changes from a conceptual idea in the early stages, to a physical entity once it is realised.

The system lifecycle phases and lifecycle processes should not be confused. A universally accepted lifecycle model does not exist, but the differences between the standards are negligible for the purposes of this article. INCOSE adopted a seven phase lifecycle model, based on the ISO/IEC15288:2008 model, as shown in

Table 1. The system lifecycle processes are invoked and iterated across and within those phases to progress the system through its lifecycle.

Table 1: Generic life-cycle stages (INCOSE, 2011)

Life-cycle stages	Purposes	Decision Gates
Exploratory research	Identify stakeholders' needs Explore ideas and technologies	Decision options: • Proceed with next stage • Proceed and respond to action items • Continue with this stage • Return to preceding stage • Put a hold on project activity • Terminate project.
Concept	Refine stakeholders' needs Explore feasible concepts Propose viable solutions	
Development	Refine system requirements Create solution description Build system Verify and validate system	
Production	Produce systems Inspect and verify	
Utilisation	Operate system to satisfy user's needs	
Support	Provide sustained system capability	
Retirement	Store, archive, or dispose of the system	

Creating and Created Systems

Closely related to the lifecycle phase of a system is the differentiation of the enterprise system that creates the product system and the resulting system. An organisation (whether a company, programme or project) exists to create a specific product, service or result (Project Management Institute, 2013). During that creation process, the architecture of the resulting system is defined. Thus, the "created system" is a result of the "creating system." This article is concerned with aligning the product and business descriptions of the "created system." This differentiation is important to ensure that effort is not expended in an attempt to align the descriptions of the "creating system" and its resulting "created system."

PRODUCT SYSTEM DESCRIPTIONS

Engineering projects typically generate large quantities of design documentation, especially when developing complicated systems involving multiple engineering disciplines. A systems engineer will usually attempt to describe the system by creating integrated views and models, focussing on the functions and performance of the system, instead of physical characteristics. Such functional architecture and logical design models may be presented together with the system requirements in documents such as system design descriptions. These design descriptions typically contain limited physical design information, but at least lists the subsystems, assemblies and components of the system. Those system elements are usually structured in a hierarchy, to show how the system decomposes into smaller elements.

System Breakdown Structure

Breakdown structures are used very widely, such as work breakdown structures and product breakdown structures. Decomposing something complicated into smaller, simpler pieces is such a natural approach for systems engineers. ISO26702:2007 (IEEE Computer Society et al., 2005) and ISO15288:2008 (IEEE Computer Society et al., 2008) present significantly different views of the system breakdown structure, causing unfortunate confusion. So much so that a mapping between the two structures is provided in Annex C of ISO26702:2007. The system breakdown structure shown in ISO26702:2007 seems more complete though, as it includes the processes that create, utilise, support and retire the product. Figure 1 shows the system breakdown structure as presented in ISO26702:2007.

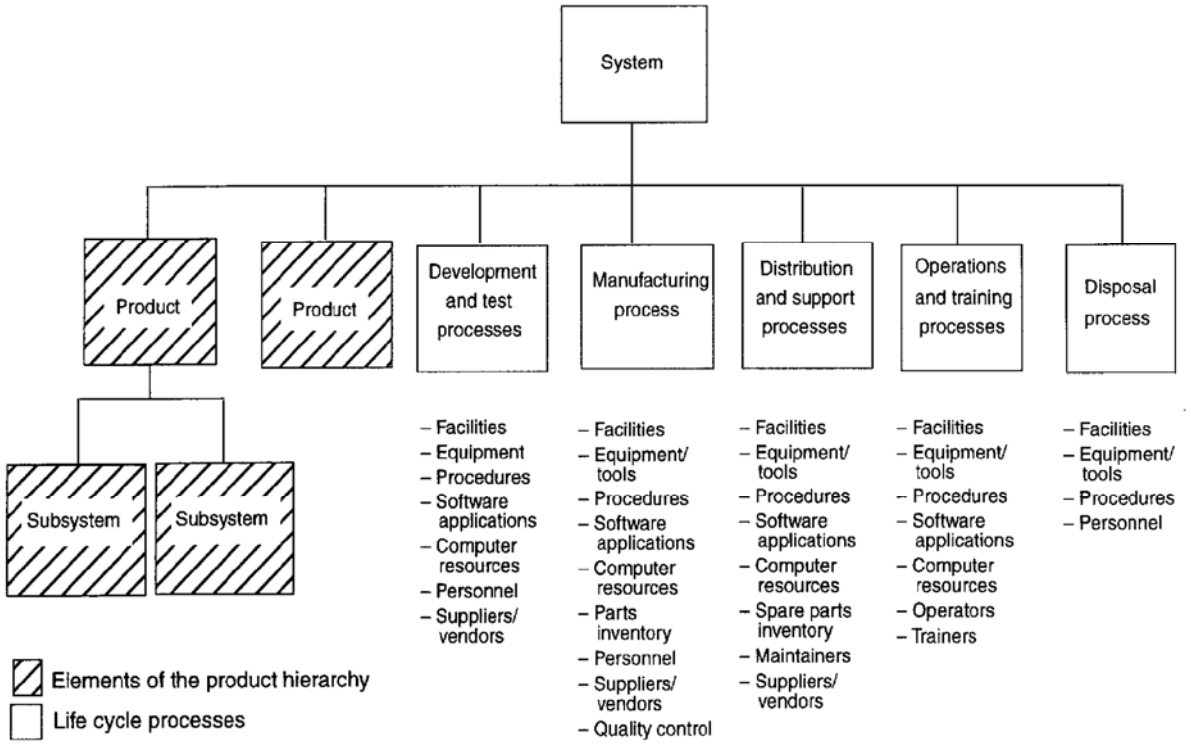

Figure 1: Generic system breakdown structure (IEEE Computer Society et al., 2005)

For many products, the processes shown in Figure 1 are performed by entire businesses. These businesses should also be designed to ensure that the product is distributed, supported and disposed according to specification and the product is utilised or operated as intended. The discipline of enterprise engineering is specifically concerned with the design and realisation of businesses.

System hierarchy levels

Engineering extensively makes use of analysis to decompose complicated systems into smaller parts (INCOSE, 2011). This allows for improved understanding of the constituent parts and how they relate to each other. It comes naturally then for engineers to describe these systems as a hierarchy, showing how the system consecutively breaks down into smaller elements that the system comprises of. However, due to the differences between industries and their respective systems, no

generally accepted and agreed system hierarchy framework exists, to be used for all engineered systems.

Nevertheless, system hierarchy levels remain a very useful way to explain the different practices and considerations applicable at the different system levels. Kossiakoff et al. (2011) uses a pyramid, as shown in Figure 2, to show how the system levels relate to each other. Boulding (1956) referred to successive system levels to explain how larger systems are composed of smaller systems. He starts at the lowest level and refers to the geography and anatomy of the universe, then works up to level nine, named the transcendental level, with systems of unknowns and unknowables.

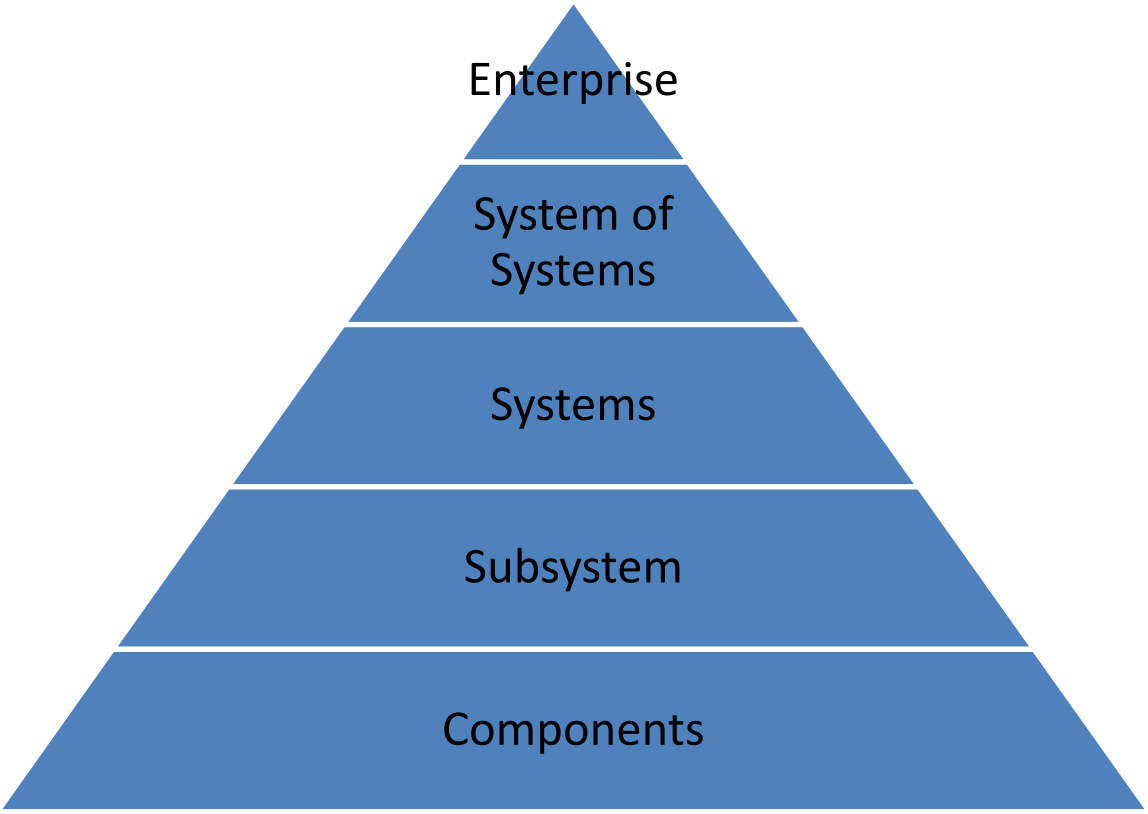

Figure 2: Pyramid of system hierarchy (Kossiakoff et al., 2011)

De Waal and Buys (2007) created a standardised system hierarchy to be used in the South African Department of Defence. Some of its terminology is defence industry specific, but the hierarchy has seen adoption in many other sectors, such as energy and transport. Figure 3 shows the ten system levels related to the Boulding hierarchy of general systems theory. The Boulding Hierarchy is only included to give additional information on the ten system levels.

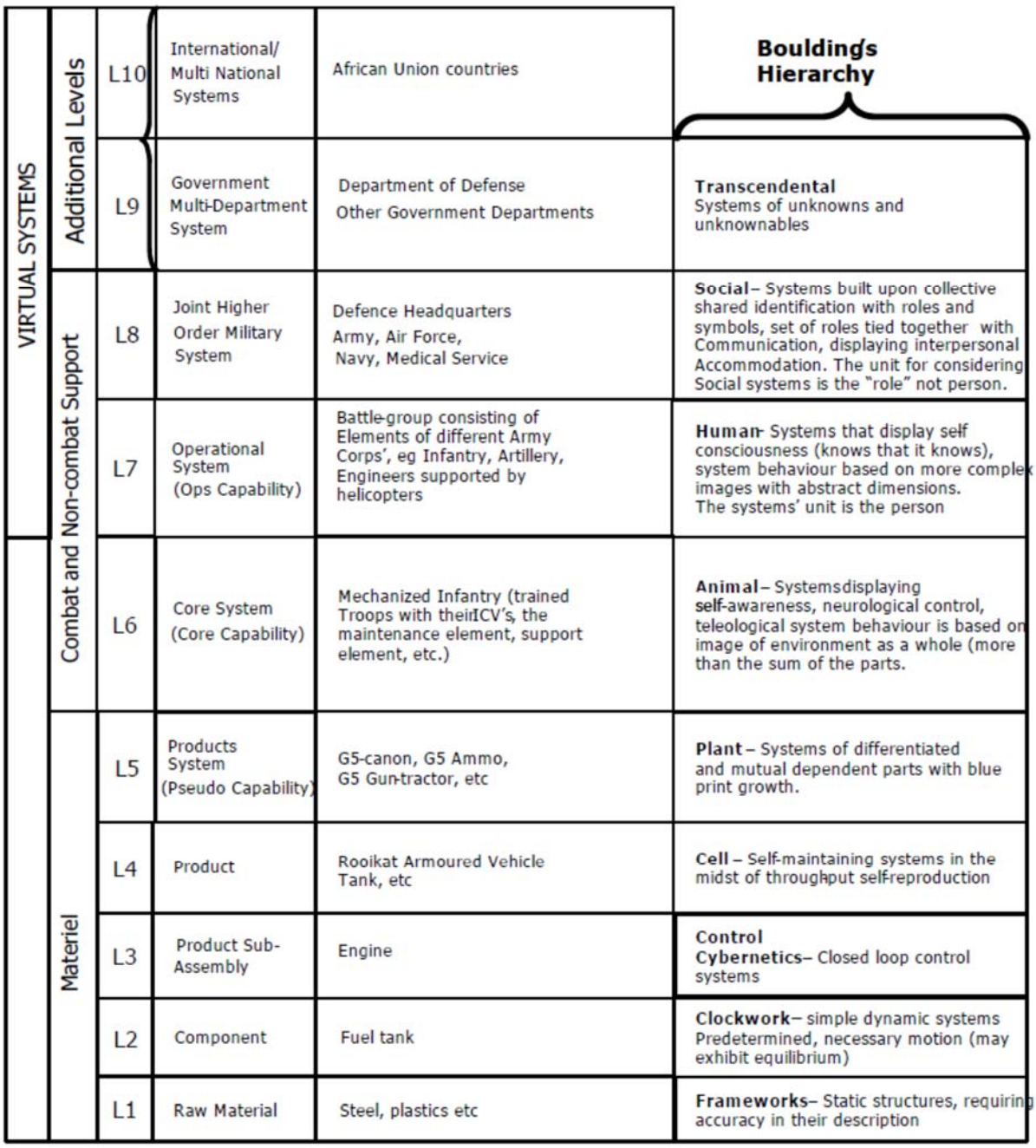

Figure 3: Ten levels of system hierarchy (De Waal and Buys, 2007)

Operational systems are found at level seven in this hierarchy and are also referred to operational capabilities. Similarly, level six is also described as a capability and level five is very specifically referred to as a pseudo capability. The significance of this differentiation is that level five systems alone are not capable of producing a desired output, because the personnel to use, operate or support the system are not considered as part of this level. At level six, the capability is available, with the system consisting of the following elements (C.J. Smith and R. Oosthuizen, 2011):

- Personnel: Characteristics of the required human resources;
- Organisation: Structure and control to direct resources;
- Sustainment: Characteristics of the logistics, personnel and financial support;

- Training: Characteristics of the training required to prepare human resources;
- Equipment: The type, quantity and characteristics of the equipment required;
- Doctrine: Characteristics of the required instructions, aids, operating procedures etc.;
- Facilities: Characteristics of the facilities required;
- Information: Characteristics of required intelligence, information and data;
- Technology: Characteristics of commercial or other technologies required.

Therefore, operational systems consist of lower level systems, each with its own purpose and characteristics. This corresponds closely to the acknowledged system of systems category, as defined by the US Department of Defence (Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, 2008). In this type of system of systems, the constituent systems retain their independent ownership, objectives and resources. Sage and Cuppan (2001) offer the following five characteristics of all systems of systems, regardless of the category:

- A system of systems is composed of systems that are independently useful;
- The component systems may be acquired and operated independent of the system of systems;
- The component systems are often geographically dispersed;
- The system of systems displays emerging behaviour that do not reside in any of its component systems;
- Systems of systems continuously evolve over time and are never complete or fully formed.

The DoD Systems of Systems Engineering document offers guidance on the engineering of systems of systems, but Kossiakoff et al. (2011) argues that the basic tools that we have in systems engineering may not be sufficient to engineer systems of systems. It has been shown though, that selected systems engineering techniques can be applied to systems of systems and enterprise systems, especially those that deal with the system as a whole, instead of its elements (Erasmus, 2014).

ENTERPRISE SYSTEM DESCRIPTIONS

Enterprise engineering as a discipline was born from the need to design enterprises as a comprehensive and coherent system, instead of allowing the continued ad-hoc emergence of enterprises (de Vries et al., 2013). Martin (1995) stated that “Enterprise Engineering is an integrated set of disciplines for building or changing an enterprise, its process and systems.” Hoogervorst (2009) argues that good enterprise design is essential for high performance and that enterprise engineering is underpinned by the fields of enterprise ontology and architecture. Ontology and architecture are certainly two familiar concepts to systems engineers and should yield areas of overlap and commonality allowing for unification of system descriptions.

Enterprise Ontology

The Open Group defines an enterprise as the highest level of organisation currently under consideration and typically includes all missions and functions (The Open Group Architecture Forum et al., 2011). Thus, enterprise in this sense is the equivalent of the system-of-interest of the current endeavour. By considering it as a system, an enterprise is composed of system elements which can roughly be categorised into people, processes or tools. The Open Group Architecture Framework (TOGAF) considers 35 element types (meta-objects), divided into architecture domains, as shown in Figure 4.

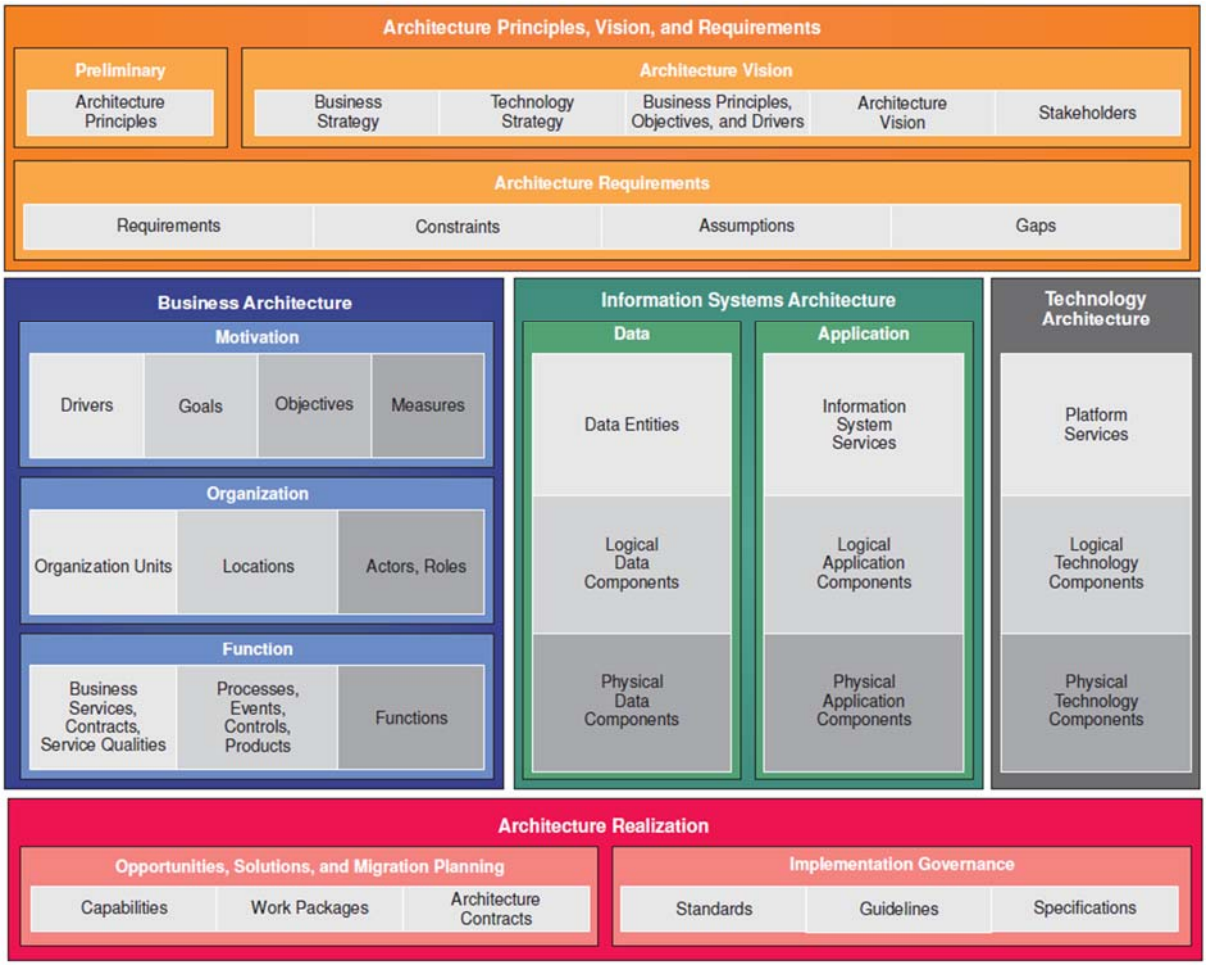

Figure 4: Representation of the TOGAF9 Content Metamodel (The Open Group Architecture Forum et al., 2011)

Even though the enterprise is composed of physical elements, the enterprise itself is not a physical entity. The elements that the system consists of, are in most cases independent, but brought together to achieve a specific goal. Therefore, an enterprise is a complex system, consisting of many interrelated elements with nonlinear relationships (Bennet, 2011). However, an enterprise consumes, produces and incorporates many other systems, making it a system of systems. Harmon (2005) argues that an enterprise is best described as an intelligent complex adaptive system of systems (ICASOS).

Enterprise Architecture

Ontology only allows for description and definition of the elements of the enterprise. By its nature though, a system's behaviour is not only a result of its components, but also emerges from the relationships between those components (Green and Bossomaier, 1993). The term "system" covers a very wide spectrum though, from tiny biological systems to galaxies, or even abstract systems such as mathematical models. Most systems share some common characteristics, including the following (Baianu, 2011):

- Systems are abstracts of reality;
- Systems have structures, defined by its parts;
- Systems have behaviour, such as the processing of material, information or energy;
- The parts of the system have functional and structural relationships.

Enterprise architecture is a way of depicting the current or desired elements of the enterprise system and the relationships between those elements. Zachman (1987) stated that such architecture descriptions are created to manage complexity and change. TOGAF provides a meta-model that shows the standard relationships between its 32 meta-objects (The Open Group Architecture Forum et al., 2011). This meta-model is the starting point for creating catalogues, matrices and models of an enterprise, to ensure that relationships between enterprise elements are captured according to the framework principles.

Enterprise Levels and Layers

To allow for improved comprehension of the architecture, many enterprise architecture frameworks are structured according to layers or domains. Such a categorisation of the meta-objects also allows for division of responsibility and specialisation of the practitioners who develop and implement those enterprise elements. The Open Group Architecture Framework meta-objects are structured according to the phases of the architecture development method, with five layers which can further be subdivided into a total of 11 sub-layers objects (The Open Group Architecture Forum et al., 2011).

By filtering the list of TOGAF meta-object relationships to only show composition and decomposition relationships, it is found that meta-objects only decompose to the same meta-object. Simply stated, none of the meta-objects are composed of other meta-objects. Business functions do not decompose to processes and services. Instead, function "supports" or "is realised by" processes. Functions only decompose to lower level functions. Thus, the only hierarchical levelling present in the standard framework, is within single meta-objects. Each architecture layer contains several hierarchies, for each of the meta-objects it is made up of.

UNIFICATION THE PRODUCT AND BUSINESS DESCRIPTIONS

Humans are the first element of the POSTEDFIT breakdown of capabilities (C.J. Smith and R. Oosthuizen, 2011). With capabilities found at level six and seven of the system hierarchy, it is clear that a business system, including its personnel, will be found at level six or higher. Figure 2 shows Enterprise as the top system level. Thus, enterprise will include the various product systems and business systems operated as part of the enterprise. With humans sitting at least at level six of the de Waal and Buys (2007) framework, it can be seen that an enterprise fits in well at level seven, as

an operational system of systems. This is a good starting point to unify the descriptions of business and product systems.

Integrated system breakdown structure

The level seven enterprise system will decompose various product and business systems, similar to what is shown in Figure 1. Figure 5 shows an example of such an integrated system breakdown structure, showing the enterprise SoS at level seven of the system hierarchy. Each of the level six elements of the enterprise is a system of systems in its own right. Those level six systems of systems will subsequently decompose into appropriate product or business elements, according to the selected framework. The product SoS may be depicted with a product breakdown structure or system architecture, whilst the utilisation or logistics support businesses may be described using enterprise architecture. Each level five element of the business systems represent a business activity area or function, with its associated processes, tools, data, rules, etc.

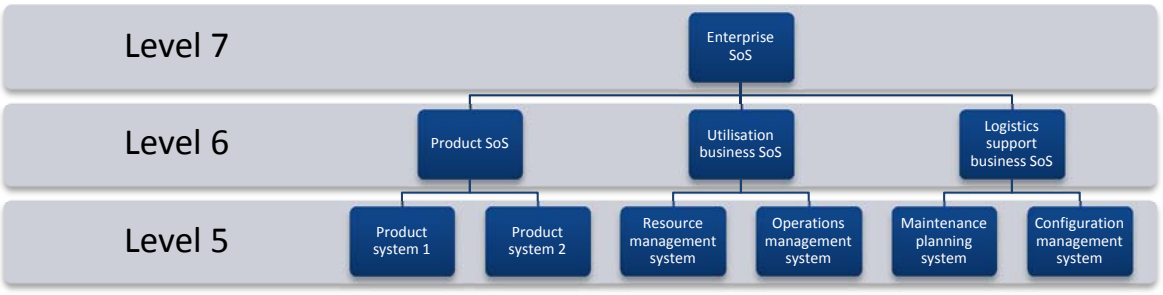

Figure 5: Integrated system breakdown structure

Such a view is easy to understand when the product system is one of the many systems used as part of a particular enterprise. When the product includes one or more complete businesses however, such as the power station example mentioned previously, it seems counter intuitive for the Product SoS to be seen as part of the enterprise. This is a matter of semantics though, only encountered when the power station, or similar complicated system, is seen as a traditional product at level five or lower. Instead, the power station should be viewed as an enterprise system of systems, which includes product systems and business systems. This enterprise SoS is created by a separate and distinct enterprise SoS. Thus, the power station is the “created system of systems.”

Enterprise system ontology

It has been well established that customers do not value a piece of hardware or software, but the ability to do something (Zeithaml, 1988; Anderson, 1995; Woodruff, 1997). Acquisition programmes do not aim to acquire a power station or an aircraft carrier, but rather a capability to generate power or transport aircraft to the theatre of war. Thus, it is the output of the product or service that is sought and valued, instead of the hardware and software that generates the output. Enterprises deliver products and services and it uses the power station or aircraft carrier to do so. Therefore, an operational enterprise system is the desired output from the engineering effort.

In fact, it is not only the delivery of a specific output that is desired and valued, but the sustained delivery of said output (Labuschagne and Brent, 2005). Blanchard presented Figure 6 to illustrate the

basic elements necessary to support a system to perform its intended function and satisfy its requirements. Evaluating those nine elements in terms of enterprise ontology allows for some mapping between a traditional system description and enterprise description.

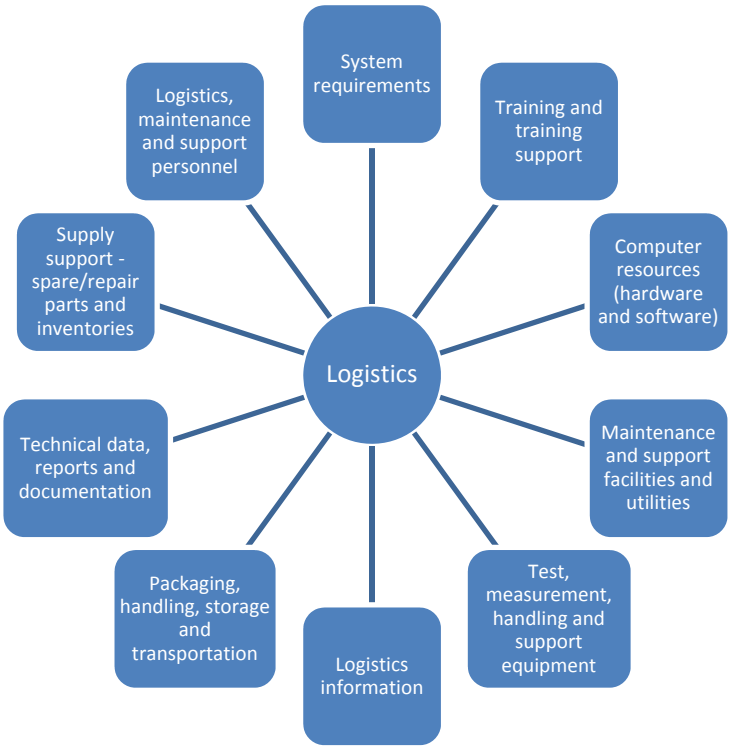

Figure 6: The basic elements of logistics, based on Blanchard (2004)

Table 2 shows a mapping between the nine logistic support elements and the most appropriate enterprise meta-objects that can be used to represent those logistic elements. Six of the logistic support elements can easily be represented as enterprise elements, but the remaining three do not have obvious mappings.

Table 2: Mapping between logistic elements and enterprise ontology

Logistic support element	Comparable enterprise element
System requirements	Requirements
Training and training support	Business process
Computer resources (hardware and software)	Technology component
Maintenance and support facilities and utilities	
Test, measurement, handling, and support equipment	
Logistics information	Data entity
Packaging, handling, storage and transportation	Business function
Technical data, reports and documentation	Data entity
Supply support - supply/repair parts and inventories	
Logistics, maintenance and support personnel	Actor

The logistic support elements which can't be represented using enterprise meta-objects indicate gaps in enterprise ontology. Enterprise architecture frameworks typically do not include meta-objects for facilities, buildings, materials and equipment which is not information technology related. Considering the central role of business processes in enterprise architecture and business modelling, it is concerning that process inputs and outputs of business processes are limited to information elements.

CONCLUSION

A common frame of reference between systems engineering and enterprise engineering is achievable. This is especially true as enterprises and products become more entwined and inter-dependant. However, unification of these two disciplines seems out of reach at this stage. The terminologies and approaches differ sufficiently to make complete unification seem unrealistic. Instead, we propose that the two disciplines remain separate and specialised in describing and defining the different types of systems. Systems architecture is proficient at defining and describing product systems, as much as enterprise architecture is for enterprise systems. What remains very useful though, is to have a common point of reference between the two disciplines. We have found this point to be at level six of the system hierarchy, where the product and business systems assimilate to create a level seven enterprise system of systems. Level seven is essentially the enterprise level and when decomposed, divides into the product or business domains.

Our original premise remains: the two disciplines can benefit each other. We have shown that improved cooperation between the two disciplines is possible when starting from the given fixed position and by applying clear transition criteria between levels. Ideally, the set of requirements should include both business and product requirements, allowing for improved cooperation and collaboration between the two disciplines. This will also allow the practitioners to identify new requirements that typically arise from the emergent properties of systems.

FUTURE WORK

We found that the current enterprise ontology did not cater for many of the objects typically considered when designing the operating and support businesses of engineered systems. We propose that this is because enterprise ontologies are typically created by empirical observation, instead of a good theoretical foundation. The next step is to create an enterprise system ontology based on theory, rather than empirical evidence.

REFERENCES

Anderson, J.C., 1995. Relationships in Business Markets: Exchange Episodes, Value Creation, and their Empirical Assessment. Journal of the Academy of Marketing Science 23, 346–350. doi:10.1177/009207039502300415

Baianu, P.D.I.C., 2011. Complex Systems Theory and Biodynamics, in: Baianu, P.D.I.C. (Ed.), Complexity, Emergent Systems and Complex Biological Systems. PediaPress: Mainz, Germany.

Bennet, A., 2011. Organizational survival in the new world: the intelligent complex adaptive system, 1st edition. ed, KMCI Press. Routledge, Amsterdam ; Boston.

Blanchard, B.S., 2004. Logistics engineering and management, Sixth Edition. ed, Prentice-Hall International Series in Industrial and Systems Engineering. Pearson Prentice Hall, Upper Saddle River, N.J.

Boulding, K.E., 1956. General Systems Theory—The Skeleton of Science. Management Science 2, 197–208. doi:10.1287/mnsc.2.3.197

C.J. Smith, R. Oosthuizen, 2011. Applying systems engineering principles towards developing defence capabilities, in: ISEM 2011 Proceedings. Presented at the The Industrial, Systems and Engineering Management conference, Stellenbosch, South Africa, p. 103.

De Vries, M., van der Merwe, A., Gerber, A., 2013. Towards an enterprise evolution contextualisation model, in: Enterprise Systems Conference (ES), 2013. Presented at the ES2013, IEEE, Cape Town, South Africa, pp. 1–12. doi:10.1109/ES.2013.6690078

De Waal, J., Buys, A.J., 2007. Interoperability and standardisation in the Department of Defence : an exploratory study. The South African Journal of Industrial Engineering 18. doi:10.7166/18-1-140

Erasmus, J., 2014. Requirements engineering for human activity systems, in: EMEASEC 2014 Proceedings. Presented at the INCOSE EMEA Sector 2014 Systems Engineering Conference, INCOSE, Cape Town, South Africa. doi:10.13140/RG.2.1.2329.8085

Erasmus, L.D., Doeben-Henisch, G., 2011. A theory for systems engineering management, in: ISEM 2011 Proceedings. Presented at the The Industrial, Systems and Engineering Management conference, Stellenbosch, South Africa.

Green, D.G., Bossomaier, T.R.J. (Eds.), 1993. Complex systems: from biology to computation. IOS Press, Amsterdam.

Harmon, K., 2005. The “systems” nature of enterprise architecture, in: Systems, Man and Cybernetics, 2005 IEEE International Conference on. Presented at the SMC2005, IEEE, Waikoloa, HI, USA, pp. 78–85 Vol. 1. doi:10.1109/ICSMC.2005.1571125

Hoogervorst, J.A.P., 2009. Enterprise Governance and Enterprise Engineering, 2009 edition. ed, The Enterprise Engineering Series. Springer Science & Business Media, Diemen, The Netherlands.

IEEE Computer Society, Software and Systems Engineering Standards Committee, Institute of Electrical and Electronics Engineers, IEEE-SA Standards Board, 2005. IEEE standard for application and management of the systems engineering process, First edition. ed. Institute of Electrical and Electronics Engineers, New York, N.Y.

IEEE Computer Society, Software & Systems Engineering Standards Committee, Institute of Electrical and Electronics Engineers, International Electrotechnical Commission, International Organization for Standardization, 2008. Systems and software engineering system life cycle processes, Edition 2. ed. ISO/IEC-IEEE, Geneva.

INCOSE, 2011. Systems engineering handbook: a guide for system life cycle processes and activities, 3.2.2 ed. International Council of Systems Engineering, San Diego, CA.

International Organization for Standardization, International Electrotechnical Commission, Institute of Electrical and Electronics Engineers, 2011. Systems and software engineering – Life cycle processes – Requirements engineering. ISO/IEC-IEEE, Geneva, Switzerland.

Katina, P., Keating, C., Jaradat, R., 2014. System requirements engineering in complex situations. Requirements Engineering 19, 45 – 62.

Kossiakoff, A., Sweet, W.N., Seymour, S.J., Biemer, S.M., 2011. Systems engineering principles and practice. Wiley-Interscience, Hoboken, N.J.

Labuschagne, C., Brent, A.C., 2005. Sustainable Project Life Cycle Management: the need to integrate life cycles in the manufacturing sector. International Journal of Project Management 23, 159–168. doi:10.1016/j.ijproman.2004.06.003

Martin, J., 1995. The great transition: using the seven disciplines of enterprise engineering to align people, technology, and strategy. AMACOM, New York.

Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, 2008. Systems Engineering Guide for Systems of Systems (Standard). US Department of Defense, Washington, DC.

Project Management Institute, 2013. A guide to the project management body of knowledge (PMBOK guide), Fifth edition. ed. Project Management Institute, Inc, Newtown Square, Pennsylvania.

Sage, A.P., Cuppan, C.D., 2001. On the Systems Engineering and Management of Systems of Systems and Federations of Systems. Information, Knowledge, Systems Management 2, 325–345.

Stark, J., 2011. Product lifecycle management: 21st century paradigm for product realisation, 2nd ed. ed, Decision engineering. Springer, London ; New York.

The Open Group Architecture Forum, Dave Hornford, Tara Paider, Chris Forde, Andrew Josey, Garr y Doherty, Cathy Fox, 2011. TOGAF Version 9.1, Open Group Standard. The Open Group, USA.

Woodruff, R.B., 1997. Customer value: The next source for competitive advantage. Journal of the Academy of Marketing Science 25, 139–153. doi:10.1007/BF02894350

Zachman, J.A., 1987. A framework for information systems architecture. IBM Systems Journal 26, 276–292. doi:10.1147/sj.263.0276

Zeithaml, V.A., 1988. Consumer Perceptions of Price, Quality, and Value: A Means-End Model and Synthesis of Evidence. Journal of Marketing 52, 2–22. doi:10.2307/1251446