The University of Bradford Institutional Repository

Link to publisher's version: *https://doi.org/10.1016/j.procir.2017.01.053*

Citation: Campean F and Yildirim U (2017) Enhanced sequence diagram for function modelling of complex systems. Procedia CIRP. 60: 273-278.

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 60 (2017) 273 – 278

www.elsevier.com/locate/procedia

27th CIRP Design 2017

Enhanced Sequence Diagram for Function Modelling of Complex Systems

Felician Campean* and Unal Yildirim

University of Bradford, Bradford, BD7 1DP, UK

* Corresponding author. Tel.: +44-1274234569. *E-mail address:* F.Campean@Bradford.ac.uk

Abstract

This paper introduces a novel method referred to as Enhanced Sequence Diagram (ESD) to support rigorous functional modelling of complex multidisciplinary systems. The ESD concept integrates an exchanges based functional requirements reasoning based on a coherent graphical schema, integrated with the system operational analysis based on a sequence diagram. The effectiveness of the method to support generic function modelling of complex multidisciplinary systems at the early conceptual design stages is discussed in conjunction with an electric vehicle powertrain example, followed by an assessment of potential impact for broader application of the method in the industry.

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference

Keywords: Systems engineering, sequence diagrams, functional requirements, complex multidisciplinary systems.

1. Introduction

The complexity of technical systems such as the automotive vehicles has increased significantly in order to address evolving customer needs and environmental concerns. From a product development process viewpoint, the introduction of new technologies and control features amplify the need for integrated function modelling of complex multidisciplinary systems with a strong focus on interactions between systems from different engineering disciplines. This challenge is further compounded by the prevalence of software based features controlling the enhanced functionality of the systems of systems.

From an engineering design perspective, many functional reasoning schemes have been proposed (see [1] for a comprehensive review). Many of the well-established schemes, including Stone and Wood [2], Ulrich and Eppinger [3] and Ullman [4], have essentially evolved from an electro-mechanical perspective, and are rooted in the Pahl & Beitz [5] flow based thinking, i.e. represent the functional model of a system in terms of the flows of material, energy and information through the system. Several other researchers (e.g. Umeda et al. [6], and Goel et al. [7]) have determined functional requirements of systems in relation to the linkage between function and the intended behavior and structure of the system. However, it has been recognized that these functional modelling schemes are somewhat limited in supporting the design analysis of complex multidisciplinary systems, which require top-down solution neutral analysis and synthesis of the architecture across multiple interconnected functional flows integrated according to the operating concept of the system across the whole lifecycle. A specific limitation of such functional models is the capture of the time domain requirements corresponding to the sequence of events associated with the system operation modes and use cases, within a “global” functional model of the system.

Systems engineering approaches have evolved from the need to address the overall integration of operational and functional models within a system-of-systems context, with strong trace-ability, inter-operability and re-use properties. In particular Model Based Systems Engineering (MBSE) makes use of formal modelling methods to support functional analysis of complex systems. System Modelling Language (SysML), which emerges as a prevalent MBSE system modelling environment, provides a graphical semi-formal descriptive “language” for system modelling, focused on capturing functional requirements from the operational concept of the system across its lifecycle use cases. Each use case can be analyzed and described in detail using tools such as sequence diagrams, activity diagrams or state machine diagrams at the

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference
doi:10.1016/j.procir.2017.01.053

designer's discretion (Friedenthal et al. [8]). Several modelling languages exploit SysML diagrams, e.g. Harmony (Hoffman [9]) and Object-oriented analysis and design (Booch et al. [10]). A strength of the MBSE framework is that it combines an abstract graphical representation of the operations and functional model of a system with a detailed simulation model (either exchanges or transactions based – e.g. using state charts or state machines, or transformations based in conjunction with physical simulation environments such as Modelica, Simulink, etc). This provides strong support for system analysis, but essentially relies on the availability of models of the system of appropriate maturity (fidelity and resolution) for the known or assumed system architecture.

The current paradigm of automotive systems development, taking for example challenges like autonomous driving, requires modelling of complex systems for largely unknown scenarios, for which models are not available. Similarly, the introduction of enhanced control features on existing systems (e.g. advanced driver assist systems – ADAS, or advanced on-board diagnostics systems for powertrain control for dynamic emissions management), require structured and systematic approaches for generic functional analysis to support "solution neutral" analysis in early concept phase, which can then be implemented in different system architectures during the analysis and design development phase. Such generic function modelling framework needs to consistently and concurrently capture: (i) the sequence based functional requirements (to implement the desired operational concept); as well as (ii) the exchanges based functional requirements (i.e. the behaviors that deliver the transactional and transformative functionality required to deliver the sequence). The MBSE framework uses tools such as activity and sequence diagrams that adequately capture the sequence of events, but does not support equally well a rigorous exchanges based abstract functional modelling based on flow based thinking in a solution neutral manner.

The research presented in this paper addresses these limitations by introducing a novel functional modelling method referred to as Enhanced Sequence Diagram (ESD). The ESD concept introduces an exchanges / flow based functional requirements reasoning and schema integrated with the system operational analysis based on a sequence diagram, to provide a rigorous generic functional modelling framework of a complex multidisciplinary system. The organization of the paper is as follows: section 2 gives an overview of the background literature on flow-based and sequence-based function modelling approaches, focusing in particular on a critique of the current approach to sequence diagrams; section 3 introduces the proposed schema and methodology for ESD, followed by an illustrative application example in section 4, based on an electric vehicle powertrain; the paper concludes with a discussion and conclusions section.

2. Critical review of flow and sequence based function modelling approaches

2.1. Flow-based function modelling

Integrated Function Modelling framework [13] follows a task-oriented approach in function modelling of a system, by representing the flows of functions through the system with respect to causality, i.e. the first function is connected to the second function, and so on. Pahl & Beitz [5] have introduced a concise taxonomy for the flows through a system in terms of energy, material and signal (information), commonly underpinning functional chains in the task-oriented approaches. The overall function of an engineered system is articulated in verb-noun format and represented in a black box with input-output flows of energy, material and signal. Function structures are created by decomposing into sub-functions, with associated black boxes describing what the elements of the system might do in order to fulfil the overall function. Sub-functions are linked with arrows denoting the flows of material, energy and information. Function chains (e.g. material flow) are created to combine into a single functional structure model, achieving the overall function. The basic principles of the function modelling approach proposed by Pahl & Beitz have been widely adopted in engineering design literature (see [13] for detail).

Functions can also be conceptualized and represented in relation to state transitions, e.g. based on the general principles of the statecharts [14]. Figure 1 illustrates a state based representation of function in which a state is denoted by a box and a function is represented by an arrow [15]. The function is defined in relation to the requirement to achieve the transfer or transformation between the input and output states of the flow or operand. A systematic approach for function modelling underpinned by a state-based representation of the flows through a complex system, introduced as a System State Flow Diagram (SSFD) by Campean et al. [15], has been further described by Yildirim [16]. The essence of the reasoning that underpins this function modelling framework is illustrated in Figure 2. This is based on the Object-Attribute-Function framework of Sickafus [17], in which a function is conceptually defined in terms of a "triad"; i.e. the System of Interest (SoI), conceptualized as an object described by its attributes with their input values, must interact with another object (or "actor", denoted as "Object 2" in Figure 2), with certain properties (attributes / values), in order to achieve the desired function - transition to the output state, described in terms of the desired attribute (output) values of the operand.

2.2. Sequence-based function modelling

Sequence diagrams are based on the Message Sequence Charts (MSC) of the Specification and Description Language (SDL) (Weilkiens [18]). The SDL provides the basis of Unified Modelling Language (UML) and thus SysML, which are widely accepted in both academia and industry [19]. In engineering applications SysML is considered as "the most popular tool for model-based development of multidisciplinary systems" [20]. Of UML interaction diagrams (i.e. sequence diagram, communication diagram, timing diagram and interaction overview diagram), SysML uses sequence diagram in the representation of scenarios of interaction for a particular functionality (use case) of a system. In addition to UML and SysML, sequence diagrams are used in other languages too. For example, Soft Domain Driven Design uses sequence diagrams in behavior modelling [11], while UML sequence diagrams are used in object-oriented analysis and design [12].

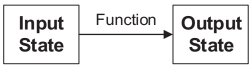

Fig. 1. State based function representation

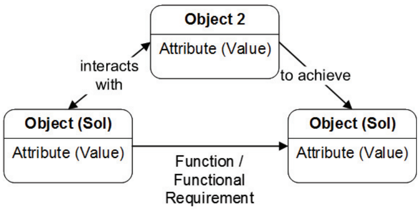

Fig. 2. SSFD Function modelling schema (SoI – System of Interest)

An actor in a SysML sequence diagram may be human or other external entity that interacts with the system under consideration, and with each other, for the fulfilment of the relevant use cases of the system. The actor and the system are represented in rectangles with dashed lines descending from the base of the rectangles with respect to time. These lines are generally termed lifelines, representing the relevant lifetime of the actor and the system. Interactions between the lifelines of the actors and the system are mapped in terms of a sequence of message exchanges, nominally in respect of time [8]. Functional requirements for the system are extracted from the sequence diagram in relation to the achievement of the exchanges of the messages. Friedenthal [8] has discussed that as well as the exchanges of messages, the passage of material and energy can also be indicated on a sequence diagram through arguments of the message. This opens the possibility for sequence diagrams to be used more widely for the analysis of multi-disciplinary systems.

Several researchers have recently proposed extensions to the use of sequence diagrams to function modelling of systems. For example, Zingel et al. [21] denoted a function by an arrow in a sequence diagram and mapped functions between the lifelines of the actors and the system in terms of concrete events for specific operation of the system. Similarly, Piques [22] complemented the sequence diagrams by functions to be implemented by the system. Figure 3 shows a sequence diagram for a hybrid vehicle based on Piques' approach. The functions are shown in terms of SysML operations (i.e. events) attached to the lifelines of the actors "driver", "road contact" and the system "hybrid vehicle".

2.3. Critique of current approach to sequence diagrams

For a purely transactional system (e.g. a communication system where all exchanges are signal or information) a sequence diagram offers an appropriate method for functional representation and functional requirements capture. However, for the analysis of multidisciplinary systems such as the hybrid vehicle system illustrated in Figure 3, based on Piques [22], the conventional sequence diagram is less useful as it does not provide sufficient information to extract meaningful exchange based functional requirements. With reference to the Hybrid Vehicle analysis in Figure 3, the sequence of activities is captured in the diagram, but with insufficient detail to extract an appropriate functional chain and the associated functional requirements. The timeline representation is somewhat irrelevant as the system operates in a dynamic mode rather than sequential, which would be more appropriately represented as a flow based functional model, mapping the transfer function of the energy flow associated with the use case illustrated.

In relation to function articulation, the interaction between the Driver and the Hybrid Vehicle as SoI, based on the Pahl & Beitz [5] taxonomy, would be classified as an information exchange, for which the functional requirement would be articulated as "import driver demand for acceleration", using the functional basis framework of Otto & Wood [23]. Clearly the exchange based functional requirements articulation is more powerful as it opens to a variety of possible solutions, which is desirable for the upfront system analysis

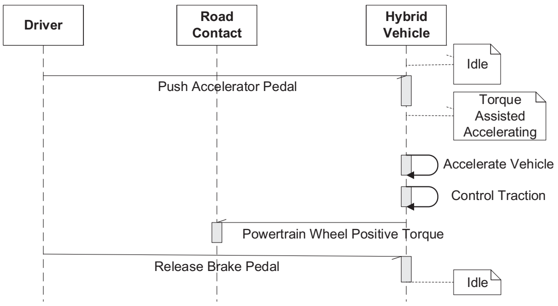

Fig. 3. Sequence diagram for a hybrid vehicle (adapted from Piques [22])

3. Enhanced Sequence Diagram

The fundamental idea behind the proposed concept of the Enhanced Sequence Diagram (ESD) is to augment the sequence diagram with flow / exchange based information such that it provides a more rigorous and information rich graphical representation of the system functions and a stronger basis for functional requirements extraction, coherent with the generic function modelling goal. In order to support this goal, the graphical representation of the ESD must embed the representation of the flows (of energy, material and information) associated with the SoI or any actor, along with the timeline. The proposed schema for the ESD is presented in Figure 4, based on the work of Yildirim [16].

The ESD diagram allows the SoI and any actor to have multiple flowlines, to represent the timeline for the events and transformations on each of the flows involved in the achievement of the functions of the system. For example, in Figure 4, the SoI is illustrated with 3 generic flowlines for Material (M), Energy (E) and Information (I). A complex system might have several flowlines of each type, which will be identified based on the Black-Box analysis of the system. The flowlines will be associated with the subsystems (i.e. internal actors in a grey box representation) that deliver the respective operations on flows. For example, a hybrid vehicle propulsion system will have an electric energy (E) flowline associated with electric energy input exchanges and transformations; it will also have a mechanical energy (E) associated with the output of torque to the axles; it will have an information flowline (I) to handle the information input and the associate transfers / transformations (e.g. electronic control system); and it will have a material flowline (M) associated with the fuel intake to the system.

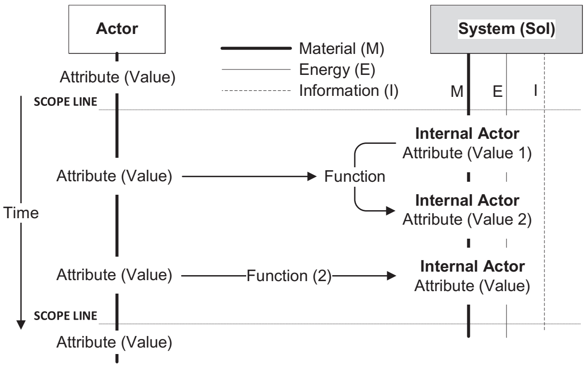

Fig. 4. Enhanced Sequence Diagram schema

The internal actors (subsystems) do not need to be identified at this stage, but their conceptualization as "objects" is required, since the flowline representation includes explicit identification of states described in terms of the parameter or attribute values. Similar to Piques [22], a function is indicated with an arrow, which in the ESD links the input state to the output state. The representation of the function is based on the SSFD modelling schema presented in Figure 2; thus, in the ESD diagram in Figure 4, this is shown with an arrow linking the external actor attribute / value box with the function unit that denotes the transition from the input state of the internal actor (value 1) to its output state (value 2). Coherent with the principle of the sequence diagrams and in particular the approach of Piques [22], the length of the flowline between the input and the output states could be taken to denote the duration of the transformation event. However, Yildirim [16] has discussed that the use of global variables of location and time in the description of the states of an object provides a more precise method, thus leaving the graphical representation to mainly indicate sequences. Transmission type functions can be represented in the ESD as illustrated with Function (2) in Figure 4, similar to the conventional statechart representation in Figure 1. The scope line in the ESD denotes the boundaries of the Use Case represented in the diagram. The initial and final values of the attribute of each actor can be indicated as states at the boundary of the scope lines.

This ESD schema applies in the same manner for interactions with external actors as well as interactions between internal actors (i.e. sequences between the internal flowlines). Given that exchanges and transformation are explicitly captured, including mapping of relevant attributes as state variables, meaningful functional requirements of the system can be clearly captured based on the ESD. The sequence of linkages between flowlines gives the functional chain associated with the achievement of the use case requirements (i.e. logical analysis of the system function), while the ESD notation supports a clear exchange based description of the functional requirements.

4. Application Example: Electric Vehicle Powertrain

To illustrate the application of the ESD, the analysis of an electric vehicle powertrain (EVP) of a full electric light commercial vehicle was considered, based on the industrial case study analysis described in [24]. Figure 5 illustrates a use case diagram for the EVP system – based on the functional analysis in [24]. For the purpose of illustration of the ESD methodology, only one use case of the EVP will be considered, i.e. the "drive mode", corresponding to the EVP function "to provide controlled torque at the rear axle" [24].

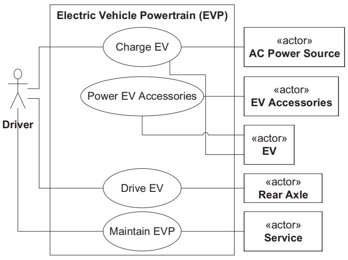

Fig. 5. Use Case Diagram for the EVP

Figure 6 shows the ESD for the "Drive EV" use case. The system of interest (SoI) is the EVP, with the Driver and the Rear Axle being the external actors that interface with the SoI.

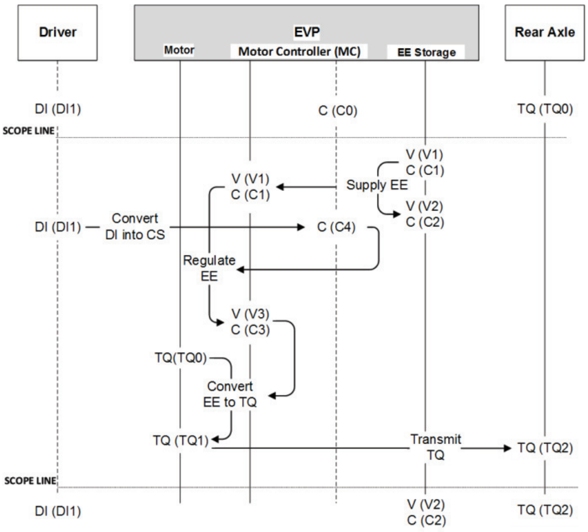

Fig. 6. Enhanced Sequence Diagram for the EVP System

The high level EVP function can be mapped across several flows: one flow of information, and three energy flows – 2 electrical and one mechanical. The internal actors, conceptualized at this stage as generic functional subsystems / objects that hold the flows, are the EE Storage subsystem, the motor controller (MC) subsystem and the motor. The functional logic and functional requirements of the EVP system at this level of analysis can be easily extracted from the ESD:

(1) Supply Electrical Energy (EE) [e.g. Voltage V1 / Current C1] from the EVP / EE Storage to the EVP / MC;

(2) Regulate EE of the EVP / MC from [V1, C1] to [V3, C3] based on control signal / acceleration request signal from the EVP / MC [C4];
(3) Convert EE [V3, C3] from the EVP / MC to Torque [TQ1] at the EVP / Motor;
(4) Transmit Torque [TQ1] from EVP / Motor to Torque [TQ2] at external actor Rear Axle; [TQ2] is the output from the system use case.

The control of the EVP is provided via function (5) to convert Driver Intent [DI1] to a control signal [C4] in the EVP / Motor Controller – shown as a transmission function / Information exchange. The functional model of the system extracted from the ESD can be shown either as a function block diagram or state based representation; Figure 7 shows a functional model (as functional block diagram) of the EVP.

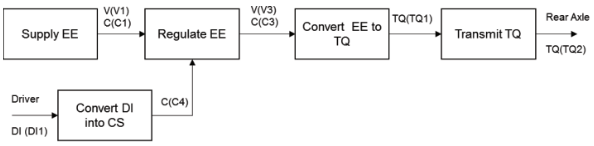

Fig. 7. EVP Functional model as a block diagram

5. Discussion and Analysis

The power of the ESD, illustrated with the Electric Vehicle Powertrain example, stems from its hybrid nature that enables to boost the sequence based functional analysis with a rigorous flow based functional modelling. Thus, in ESD the timeline of an actor is replaced with flowlines which enable a clear mapping of transactions and transformations in terms of the measurable attributes of the actor. This supports a clearer specification of the functional requirements – on an exchanges basis between interacting actors, documenting the flows both in terms of input / output, or from / to, as well as the measurable attributes that are associated with the function. The semantic articulation of the functional requirements is also much clearer (either based on the functional basis taxonomy of Stone and Wood [2], or using "shall" statements), and can feed directly into a formal requirements database. This is a significant enhancement compared to existing sequence diagram, where functions are commonly articulated in informal activity related language – as shown in the hybrid vehicle example in Figure 3, which often lead to improper articulation of functional requirements, as often observed in industrial practice.

To some extent the ESD offers flexibility to the analyst in terms of the mapping and description of the flows as functional models. Graphically, this is enabled by the 2 different ways of representing functions shown in Figure 4 – allowing both transmission and transformation types to be represented. Mapping of the flows could be performed either from the input to the output (as shown in Figure 6), or starting with the output and working out the flow requirements to the input. This can be reflected in the way functional requirements are articulated; e.g. function (4) could have been articulated as "Import Torque [TQ2] to external actor Rear Axle from EVP / Motor [TQ1]", and function (1) as "Import EE [V1, C1] from EE Storage to EVP / MC". Similarly, function (5) could have been articulated as "Import Driver Intent [DI1] as control signal"; however, this would have implied that the responsibility of converting the driver intent (which could be in the form of effort / force on a pedal as well as push button or touch screen) had already been allocated to the EVP / MC. In fact this function could be allocated to an external actor – e.g. the EV driver interface, which has not yet been defined in the system architecture.

Arguably, both ways of articulating functional requirements are correct, providing the attributes are correctly identified. In both cases the state transformations are coherently indicated in the ESD; e.g. the state of the rear axle changes from [TQ0] at the use case input (shown on the scope line) to [TQ2].

Another useful feature of the ESD is that it enables the extraction of complete functional models (as shown in Figure 7) and theoretical transfer functions with coherent mapping of parameters from the sequence diagram analysis. A model like the one shown in Figure 7 could be straightforwardly transferred to a simulation environment on the basis of physical transfer functions.

The practical importance of the approach, as illustrated with this case, is that the model has been derived on a solution neutral basis – starting with activity mapping at the use case level, and with no assumption about the architecture of the system to start off with. It has been observed that in industrial practice phenomenological models are often established first, and system models (e.g. using SysML) are derived based on the functional logic and architecture inferred from this existing model.

It is also important to note that the ESD provides a strong support for the architecture development of the system, which is very important at the early stage conceptual analysis of a new system or feature. The schema for state flow (SSFD) based modelling of functions [16], illustrated in Figure 2, requires the analyst to describe the flows of energy, material and information in terms of the state of the actor (described through its attributes). Hence, mapping the sequence of transformations required to achieve the overall function, also requires to identify internal actors, conceptualized as objects that hold the flows. Thus, the internal architecture of the system takes shape as the analysis of the functional sequence is completed. This is illustrated in the EVP example shown in Figure 6: the internal actors identified are not yet design entities, as there is no commitment to a particular design solution. E.g. the "EE Storage" actor could be any type of battery or capacitor – or indeed any combination of storage systems. Similarly, the "Motor Controller" is conceived at this stage as a nominal subsystem that – through the ESD analysis, has been allocated 2 functional requirements – (2) and (5). This shows that the ESD can also address the issue of allocation of functional requirements to internal actors / subsystems, and enables grey box analysis to be seamlessly carried out, leading to concept analysis / selections and design synthesis for each of the functional subsystems (internal actors in Figure 6).

The ESD also addresses a weakness of common flow-based functional reasoning schemes and models (discussed in Section 2 of the paper), for which there is no natural way of including time as a model variable. Within the ESD, time is naturally captured through the sequence of events, also facilitating the capture of requirements specification for duration / timing, which can be then extracted / added to the state model, as discussed in [16].

From a practical implementation viewpoint ESD has the advantage that being essentially a sequence based framework, it is easy to apply directly on a use case description – e.g. through activities mapping / capture. Adding the rigour of flow based exchanges articulation is in this context a manageable challenge, compared with most common functional modelling frameworks, where identification and mapping of flows is hard to manage on a solution independent basis. The flow based function modelling frameworks are reasonably easy to deploy in "re-engineering" mode (i.e. when the architecture and design of the system is known, and physical models are available), but notoriously difficult to implement when there is no current system (which is often the case in the automotive industry – e.g. when innovative customer experience enhancing features are introduced).

The ESD can be used at different levels of the system. The EVP case study illustrated this by representing Rear Axle at a high level, based on one flow line, while the EVP (SoI) are analysed in detail by using multiple flowlines. The EVP analysis could be further developed in parallel with the design architecture development, e.g. by representing the way the electrical energy is stored through including multiple actors and functions. This can be represented straightforwardly in the ESD, but the analysis can be expected to grow substantially in application to real complex systems. While several industrial case studies of the application of the ESD within the automotive sector have been described in [16], which provided good preliminary empirical validation of the effectiveness of the methodology, further validation is needed to prove the scalability of the approach in conjunction with large complex multidisciplinary systems.

6. Conclusions and Further Work

This paper has introduced an Enhanced Sequence Diagram as a novel method for the functional modelling of complex multidisciplinary systems. The ESD concept introduces an exchanges based functional requirements reasoning and schema underpinned by the state flow model, integrated with the system operational analysis based on a sequence diagram. The ESD ability to combine transactions and transformations type functions in the same schema is very important as it enables rigorous requirements analysis and modelling of multidisciplinary systems in a top-down solution independent manner. The paper has discussed the validation of the approach with a case study, and has reflected broadly on the applicability and impact on the approach for industrial applications.

Future work will focus on the development of a software-implementation of the ESD to assist the effective deployment to the analysis of complex systems, as well as proving the scalability of the method to modelling of large systems and in other industrial sectors – e.g. aerospace, software, and manufacturing process modelling.

References

[1] Erden, M.S., Komoto, H., Van Beek, T.J., D`Amelio, V., Echavarria, E. and Tomiyama, T. A review of function modeling: approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 2008, 22:147-169.

[2] Stone, R.B. and Wood, K.L. Development of a Functional Basis for design. *Journal of Mechanical Design* 2000, 122:359-370.

[3] Ulrich, K.T. & Eppinger, S.D. *Product design and development.* 3rd edn. New York: McGraw-Hill/Irwin; 2003.

[4] Ullman, D.G. *The mechanical design process.* 4th edn. New York: McGraw-Hill; 2010.

[5] Pahl, G., Beitz, W., Feldhusen, J. & Grote, K.H. *Engineering Design: a systematic approach.* 3rd edn. London: Springer; 2007

[6] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. and Tomiyama, T. Supporting conceptual design based on the function-behaviour-state modeller. *AI EDAM* 1996, 10(4): 275-288.

[7] Goel, A., Rugaber, S. and Vattam, S. Structure, behaviour, and function of complex systems: the structure, behaviour, and function modelling language. *AI EDAM* 2009, 23:23-35.

[8] Friedenthal, S., Moore, A. and Steiner, R. *A practical guide to SysML: The Systems Modeling Language.* 3rd edition. Burlington:Morgan Kaufmann; 2014.

[9] Hoffmann, H-P. *Systems engineering best practices with the rational solution for systems and software engineering.* IBM Corporation; 2011.

[10] Booch, G., Maksimchuk, R.A., Engle, M.W., Young, B.J., Conallen, J. and Houston, K.A. *Object-oriented analysis and design with applications.* 3rd edn. Boston: Addison-Wesley; 2007.

[11] Salahat, M. and Wade, S. *Domain Driven Design vs Soft Domain Driven Design Frameworks.* International Journal of Computer, Electrical, Automation, Control and Information Engineering. 10 (7):1330-1336, 2016

[12] Srimathi, H., Sriram, H. and Krishnamoorthy, A. *Object Oriented Analysis & Design Using UML.* Scitech; 2015

[13] Eisenbart, B. *Supporting interdisciplinary system development through integrated function modelling.* PhD dissertation. University of Luxembourg, 2014.

[14] Harel, D. (1987) Statecharts: a visual formalism for complex systems. *Science of Computer Programming* 1987; 8:231-274.

[15] Campean, F., Henshall, E., Yildirim, U., Uddin, A. and Williams, H. A structured approach for function based decomposition of complex multi-disciplinary automotive systems. *Smart product engineering: the 23th CIRP design conference.* Bochum, Germany; 2013.

[16] Yildirim, U. *Function Modelling of Complex Multidisciplinary Systems:Development of a System State Flow Diagram Methodology for Function Decomposition of Complex Multidisciplinary Systems.* PhD thesis. University of Bradford, 2016.

[17] Sickafus, E.N. *Unified Structured Inventive Thinking: How to Invent.* Michigan: Ntelleck; 1997.

[18] Weilkiens, T. *Systems Engineering with SysML/UML: Modelling, Analysis, Design*. Burlington: Morgan Kaufmann; 2006

[19] Xie, S., Kraemer, E., Stirewalt, R.E.K., Dillon, L.K. and Fleming, S.D. Design and evaluation of extensions to UML sequence diagrams for modeling multithreaded interactions. *Information Visualization* 2009; 8(2):120-136.

[20] Albers, A. and Zingel, C. Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML. *In: Proc. Of Smart product engineering: the 23th CIRP design conference.* Bochum, Germany; 2013. p. 83-92.

[21] Zingel, C., Albers, A., Matthiesen, M. and Maletz, M. Experiences and advancements from one year of explorative application of an integrated model-based development technique using C&C²-A in SysML. *International Journal of Computer Science* 2012; 39 (2):165-181.

[22] Piques, J-D SysML for embedded automotive systems: SysCARS methodology. *Embedded Real Time Software and Systems (ERTS) conference.* Toulouse, France; 2014.

[23] Otto, K. and Wood, K. *Product design: techniques in reverse engineering and new product development.* New Jersey: Prentice Hall; 2001.

[24] Campean, I.F., Henshall, E., Brunson, D., Day, A., McLellan, R., &Hartley, J. *A structured approach for function analysis of complex automotive systems.* SAE Int. J. Mater. Manuf. 4(1):1255-1267, 2011.