The University of Bradford Institutional Repository

Link to publisher's version: *https://doi.org/10.21278/IDC.2018.0543*

Citation: Campean F, Yildirim U and Henshall E (2018) Synthesis of functional models from use cases using the system state flow diagram: A nested systems approach.15th International Design Conference (Proceedings) - Design 2018. 21-24 May 2018. Dubrovnik, Croatia.

INTERNATIONAL DESIGN CONFERENCE - DESIGN 2018
Dubrovnik - Croatia, May 21-24, 2018.

SYNTHESIS OF FUNCTIONAL MODELS FROM USE CASES USING THE SYSTEM STATE FLOW DIAGRAM: A NESTED SYSTEMS APPROACH

Felician Campean, Unal Yildirim and Ed Henshall

Abstract

The research presented in this paper addresses the challenge of developing functional models for complex systems that have multiple modes of operation or use cases. An industrial case study of an electric vehicle is used to illustrate the proposed methodology, which is based on a systematic modelling of functions through nested systems using the system state flow diagram (SSFD) method. The paper discusses the use of SSFD parameter based state definition to identify physical and logical conditions for joining function models, and the use of heuristics to construct complex function models.

1. Introduction

The complexity of technical systems such as automotive vehicles has increased significantly with the introduction of new technologies to address evolving customer needs/trends such as advanced driver assist systems (ADAS) and environmental concerns including stringent CO2 requirements (Toepfer and Naumann, 2016). New technologies employed to deliver enhanced user interaction and system lifecycle management increasingly rely on cybertronic systems introduced within the legacy architecture of physical systems (Eigner et al., 2016). This brings important integration challenges (Hehenberger et al, 2016; Tomiyama et al., 2007), in particular associated with the need to identify interactions between systems from different engineering disciplines early in the engineering design process (Van Beek and Tomiyama, 2009; Lindemann et al., 2009). Addressing these challenges within a system-of-systems engineering approach places greater emphasis on the development of generic, top-down, solution independent models of complex systems that facilitate function decomposition across the levels of abstraction of the system.

The specific challenge addressed in this paper relates to the fact that from a user and/or use perspective, complex multidisciplinary systems commonly have multiple operation modes and use cases, with different functional requirements, often pertaining to different disciplines (Liu et at, 2015). Functional modelling and analysis for such contexts is often approached in practice through an independent analysis of each mode of operation / use case. This facilitates the capture of functional requirements and specifications, which feed directly into the architecture design analysis and synthesis, without attempting to integrate towards a single overall functional model of the complex system.

The well-established engineering design functional representation models (e.g. Pahl et al, 2007), rooted in the conventional electro-mechanical domain, have not been specifically concerned with the multi-disciplinary synthesis of functional models for complex systems. The Integrated Function Modelling (IFM) approach of Eisenbart (2017) promotes a concurrent approach and environment for the consideration of multiple viewpoints for the analysis of complex systems. The multi-domain structure matrix underpinning to the IFM framework provides an opportunity to consider multiple operating modes and use cases along with the user interaction / process view and function modelling. However, it is not immediately conducive to the development of concise architecture independent functional models.

The systems engineering approach (see for example Friedenthal et al, 2008, and Wilkiens, 2008) support the capture of (solution independent) requirements from multiple use cases, without necessarily attempting to build a rigorous multi-disciplinary functional model. Functional models are typically focussed towards capturing the logic of the system (as, for example, in a state machine model) which can be checked / verified / validated before the architecture / structure of the system is established. Hybrid modelling approaches is an emerging research field that aims to extend the application of logic modelling methods to continuous physical systems. These approaches are currently limited by their architecture dependence through the use of physical simulation models; the need to evolve towards solution independent functional modelling has been recognised (see for example Huang et al, 2016).

The approach behind the work presented in this paper is underpinned by the reference model of a complex system provided by Crilly (2015) which promotes the idea of multiple nested systems and the propagation of roles and functions through the nested systems (Crilly, 2013). Figure 1 provides an illustration of Crilly's nested systems concept and propagation of functions that supports the fulfilment of multiple roles in conjunction with other systems - all nested in a supersystem.

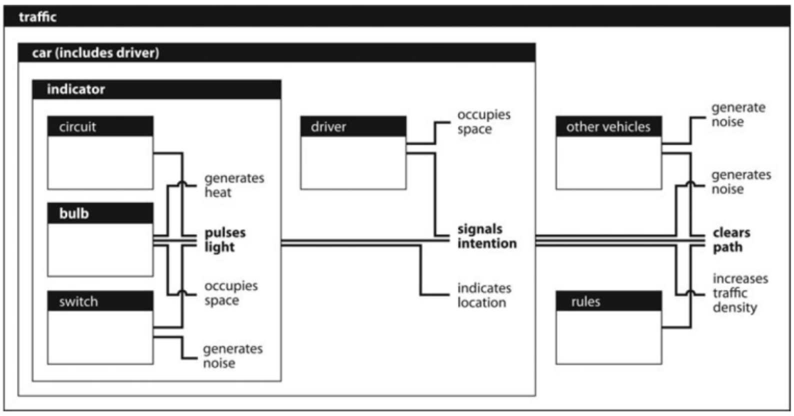


Figure 1. Illustration of function propagation through nested systems (Crilly, 2015)

The research presented in this paper focuses on the development of a methodology that supports a systematic approach for the top-down development of a function model for a complex system with multiple use cases as modes of operation of the whole system, which are not necessarily related to the same transformative flow or function chain. The System State Flow Diagram (SSFD) (Yildirim et al, 2017) is used as the reference function modelling methodology; this paper provides further insights for the SSFD based function modelling, emphasizing the strength of this framework for supporting top-down function modelling of complex multidisciplinary systems.

The consideration of operation modes / use cases structure of systems analysis in this paper introduces new perspectives to the architecture-based concept of nesting of function models discussed by Crilly (2013 and 2015). In this paper we consider three types of nesting of functional models in relation to the analysis of a complex system with multiple levels of decomposition, as follows:

- Function modelling nested within the use cases of the system;
- Function modelling nested within the architecture / subsystems structure of a system;
- Nested decomposition of functions - i.e. identification of function structures (chains, sequences) as way of achievement of a function at a lower level model.

A case study based approach is used to illustrate both the problem of function modelling of systems with multiple use cases / operating modes, and the deployment of the methodology for the integrated nested function modelling based on the SSFD.

The organization of the paper is as follows: section 2 provides an overview of the background literature on, and approaches to, function modelling; section 3 introduces the automotive case study of an electric vehicle used as the basis of analysis , and describes the application of the SSFD methodology for nested function modelling; the paper concludes with reflective discussion on the methodology and conclusion.

2. Related Work

2.1. Function modelling

Many of the well-established approaches to function modelling are based on the approach taken by Pahl & Beitz (2007) of developing functional models of a system on the basis of the flows of material, energy and information through the system structure. For example, the approaches of Stone and Wood (2000), Ulrich and Eppinger (2003) and Ullman (2010) are rooted in the Pahl & Beitz flow based thinking. Stone and Wood (2000) represent the flows for the customer requirements on a black box showing the overall function with input-output flows, as depicted in Figure 2. A chain of sub-functions is created in terms of black boxes for each input flow on the overall function by thinking of each operation of the flow through the system (Otto and Wood, 2001). Each function chain represents a functional model of the relevant structure of the system. Lastly, all of the function chains are aggregated into a single functional model. This may require the addition of new sub-functions to connect the chains together.

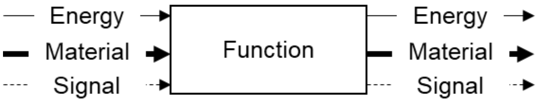


Figure 2. Black box with input-output flows

Functional modelling across multiple levels of the system has been approached in different ways. The fractal character of the Contact and Channel approach (C&C²-A) (Matthiesen and Ruckpaul, 2012; Albers and Wintergerst, 2014) promotes the analysis of existing systems with multiple functionalities at different levels of detail, but at only at a given point of time. Matthiesen and Ruckpaul (2012) emphasise that not all functions of a system are fulfilled at the same time in the majority of cases. Functions are defined in relation to a sequence that consists of at least two states corresponding to the operational modes of a system. A state can reflect different sequences and therefore can address the fulfilment of multiple functions.

The integrated function modelling approach (Eisenbart et al., 2017) represents the functional model of a system at high level through six connected views: use case view, state view, interaction view, actor view, effect view, and process flow view. The use case view lists the different use cases associated with the system under consideration. The involvement of individual processes within these use cases is represented through the process flow view. One process flow view represents the sequence of processes required for the fulfilment of one use case. A coherent graphical format of the complete process flow representation of a system with multiple use cases has not been presented yet.

The Object-Process Methodology (OPM) of Dori (2016) is capable of representing multiple functions of a system and structures that accomplish these functions in the same diagram. The decomposition of a system's function and structure can be represented through its complexity management mechanisms: unfolding/folding, in-zooming/out-zooming, and state expressing/suppressing (Osorio et al., 2011). System functions are not represented as a functional chain (a set of connected functions that capture the function structure at a given hierarchical level); therefore, it is difficult to get an integrated functional model at a consistent level of detail.

2.2. Function Modelling using the System State Flow Diagram

The System State Flow Diagram (SSFD) approach was introduced (Campean et al., 2011; Yildirim et al., 2017) as a systematic approach for function modelling underpinned by a state-based representation of the flows through a complex system. The essence of the reasoning that underpins the SSFD function modelling framework is rooted in the Object-Attribute-Function framework of Sickafus (1997), and is based on an object "triad", illustrated in Figure 3. The SSFD functional analysis starts with defining the Input and the Output states associated with the goals of the system of interest (SoI), defined from an environment centric perspective (i.e. the Input and Output states belong to the environment of the system, and are external to the SoI). The definition of the states is based around conceptualised objects that are described through measurable attributes or properties. The SSFD operational definition of the function is in relation to the state "flow" transition or transformation from the Input state to the Output state. In order to achieve the function, the Input object must interact with another object, to facilitate the transition to the Output state. Thus, the SSFD modelling representation integrates the environment centric view (represented by the Input and Output objects) with a device centric view, represented by the conceptual SoI design solution (DS), which completes the object triad. The conceptual DS is considered as an abstract object that embeds the behaviour (reflected by the object attributes and values) necessary for the achievement of the function.

The SSFD function modelling development is based on systematic iterative decomposition of the flows that underpin the required behaviour, enabling functional models at successive levels of abstraction to be developed. The SSFD function decomposition aims to identify a functional chain (as a sequence of state transitions) that explain "the way of achievement" of the behaviour associated with the delivery of the higher level function. As discussed by Kitamura & Mizoguchi (2003), thinking of the way of achievement involves reasoning about the relevant physical phenomena. If different fundamental ways of achievement of a function can be employed, then this requires consideration of alternative design concepts and architectures before the decomposition can be taken further. The global environment variables of location and time are used in the SSFD description of the states as global attributes to expedite the modelling of time and location related transitions.

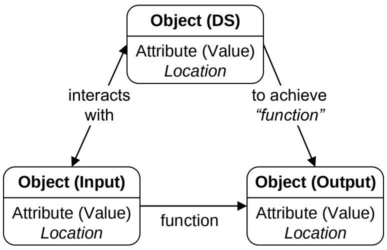


Figure 3. System State Flow Diagram function modelling schema

In order to facilitate the systematic function analysis and decomposition of a system using the SSFD function modelling, Yildirim et al. (2017) have introduced a set of process heuristics as explicit, prescriptive guidelines (Maier & Rechtin, 2000), that are similar to those introduced by Otto & Wood (2001). The 4 heuristics introduced are summarised below:

1. Main Flow Heuristic: identifies the main flow through a system in relation to the main purpose or goal of a system; the operational definition of the input and output states (i.e. objects described by attributes) at the black box level analysis of the system of interest support the identification of the main flow;
2. Connecting Flow Heuristic: defines flows of additional resources that need to be "connected" to the main flow to support the achievement of the functions on the main / higher level flow;
3. Branching Flow Heuristic: to deal with the need to introduce multiple flows associated with multiple functions related to the same input state;
4. Conditional Fork Node Heuristic: indicates a flow operation - either connecting type or branching type, which is subject to a logical or parametric condition.

The primary purpose of introducing the SSFD function modelling approach was to support the top-down solution neutral functional modelling of complex multidisciplinary systems. The structure and rigour in the definition of states supports a robust specification of requirements - in a solution neutral manner, thus supporting the development of formal state-based models earlier in the design process. This will be illustrated in the following section in conjunction with a case study.

3. Case Study: Function Modelling of an Electric Vehicle Powertrain

To illustrate the use of the SSFD for function modelling of complex systems with multiple operating modes / use cases, a case study of an electric vehicle powertrain (EVP) for a full electric light commercial vehicle is considered. This is based on an industrial case study originally presented by the authors (Campean et al., 2011), which is re-analysed using the methodology presented in this paper. The background to this industrial case study is that an electric based powertrain was to be designed and installed to replace an original Diesel based powertrain. This introduces some further challenges for the

EVP system design related to the legacy system architecture; for example, the fact that the electrified powertrain is also required provide a power supply for the low voltage vehicle accessories (such as the heating, ventilation and air conditioning (HVAC) subsystem, or the infotainment system). Figure 4 illustrates a use case diagram depicting 2 operating modes of the electric vehicle (EV): (i) "Charge EV"; and (ii) "Drive EV". The diagram also identifies three use cases directly associated with the EVP:

- UC1 – Charge EV - charge the EVP internal energy storage system;
- UC2 – Generate propulsion torque, transmitted to the external actor Rear Axle – to fulfil the user goal to "Drive EV"; and
- UC3 – to provide low voltage (LV/DC) to the vehicle accessories (i.e. Power EV Accessories) - to fulfil vehicle and user requirements for power on board.

The methodology for the case study analysis includes 3 steps:

1. Nested function modelling based on use cases and synthesis of a function model for the whole system: we will illustrate the use of SSFD heuristics to develop function models for each use case independently; these will then be aggregated into an overall system model, based on logical and parametric conditions.
2. Nested function decomposition: we will illustrate how function decomposition can be carried out within the SSFD framework to increased levels of granularity.
3. Integration of nested function decomposition analysis with the system design architecture (structure based nesting of function models): we will show the linkage between the functional and structural / design domain analysis through nested function decomposition at successive system levels.

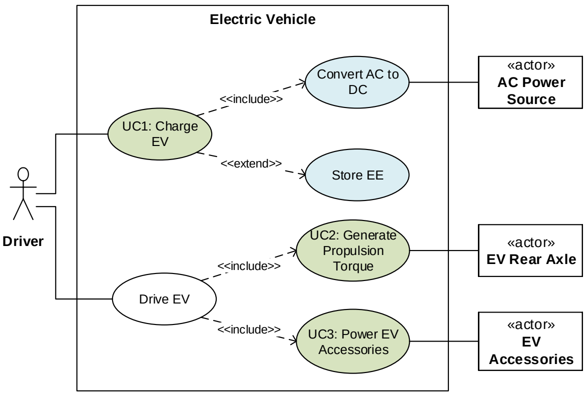


Figure 4. Use case diagram for the electric vehicle powertrain

3.1. Nested function modelling based on use cases

The common approach to the development of a functional model for a system with multiple operation modes or independent use cases is to develop a functional model for each of the use cases independently. Figure 5 illustrates a high level SSFD functional model for each of the 3 use cases in Figure 4.
The SSFD diagrams illustrate the main flow heuristic for each use case – in this case the energy flows associated with the EVP. Formal functional requirement statements for each use case are captured and illustrated in the boxes shown underneath the SSFDs. This is a high level system analysis that only shows the main transformative functions that are required in order to achieve the goal of each use case. The SSFD state representation captures the parameters (attributes) associated with the external actors and the internal states in relation to the achievement of the function(s) associated with each use case. This provides both a strong basis for modelling and the robust capture of requirements (in parametric terms), as well as a comprehensive articulation of the function – as illustrated. The SSFDs in Figure 5 also illustrate the use of connecting flow heuristics, to connect the control signal associated with the driver (user) intent, to enable the fulfilment of the functions associated with each use case. At this high level of analysis, the control signal flow (shown with dashed lines in Figure 5) is linked to the function with which it is associated, in the sense that it provides the logical condition for the fulfilment of the function. For example, the function "Convert AC to DC" should only happen when the driver input state

parameter has the value DI1. Similarly, “Generate Propulsion Torque” should only happen when the driver input state parameter takes the value DI2, and “Power EV Accessories” should only happen when the driver input state parameter takes the value DI3. The EVP “system boundary” is also indicated in the diagrams to clearly identify the internal observable states and functions of the EVP.

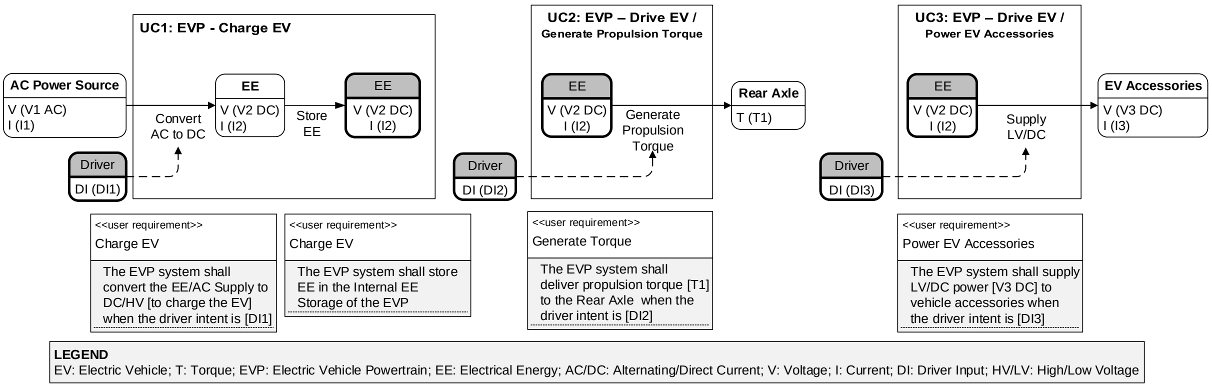


Figure 5. Function analysis for EVP Use Cases

The three SSFD function models corresponding to the three use cases are “nested” within the EVP system. A single, combined SSFD representation of the whole system model integrating the transformative and transmission functions of the three use cases can be constructed by using the SSFD heuristics, based on identification of the state couplings between the individual models. Examination of the states in the three SSFD models in Figure 5 facilitates this analysis:

- The Output State of UC1 (as EE/DC (V2, I2), in the Internal EE Storage of the EV), is the Input State for both UC2 and UC3 functions (states highlighted in Figure 5), and thus it provides a connection point for the three use case SSFDs. As use cases UC2 and UC3 can be controlled independently and could occur simultaneously during the “drive EV” operating mode (e.g. the user would use vehicle accessories - such as those associated with HVAC or infotainment, while requesting a torque demand – part of the driving task). This means that the stored energy branches into two energy flows associated with the propulsion function (UC2) and low voltage power supply to the EV accessories (UC3). Therefore, the branching flow heuristic was used to capture these flows in an EVP system SSFD model.
- The logic of the activation of the energy flows is driven by the control signal associated with the Driver Intent external input / state (highlighted in Figure 5). Such logical / parametric conditions can be represented in the SSFD using the Fork node heuristic.

Figure 6 shows the EVP system level SSFD, as a compact and succinct functional model of the system that combines the 3 separate SSFD function models corresponding to the three use cases considered. The way that the user (driver) interacts with the system can be followed through the logic of the conditions specified for the fork node; in addition to the driver intent conditions illustrated in Figure 6, additional logical constraints / specifications can be added. This could include the fact that the “Drive EV / Generate Propulsion Torque” mode cannot be selected while the vehicle is being charged, if the chosen design solution is to use a fixed charging point. However, as depicted, the analysis in Figure 6 is strictly solution neutral, by, for example, applying equally to a cabled or wireless supply of electrical power supply from the AC Power Source. The parameters of the states of the energy flow support the high level modelling of the system via analytical transfer functions.

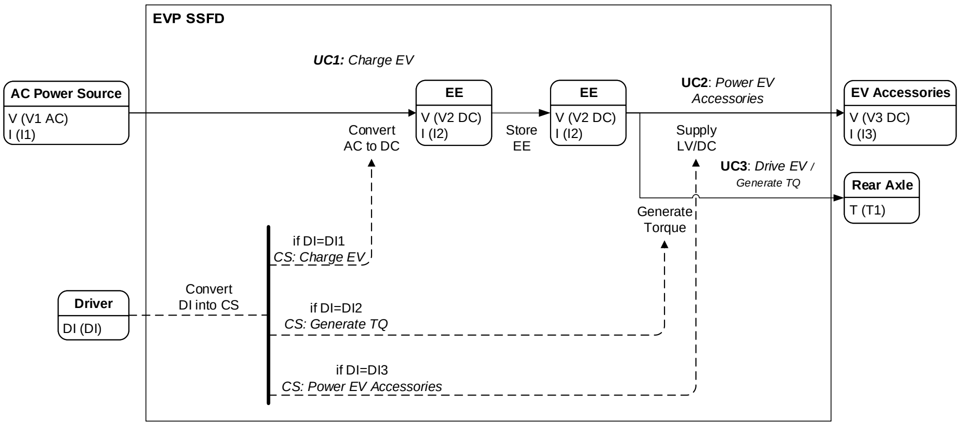


Figure 6. Integrated EVP SSFD system function model - including all use cases

3.2. Nested Function Decomposition

The system function model in Figure 6 can be further decomposed to generate a more detailed functional model for a lower level of abstraction in the systems engineering cascade. This is often a desired step in order to facilitate optimal system architecture analysis. Within the SSFD modelling framework, function decomposition involves identification of intermediate states in the flow between the Input and Output, which should be defined as observable states of objects (in terms of their measurable attributes), in the same manner as the input and output states. Thus, a functional chain (as a sequence of state transitions) is defined, which explain "the way of achievement" of the higher level function, which is expanded / replaced in the lower level SSFD.

Figure 7 shows a nested function decomposition for EVP system functional model presented in Figure 6, which provide higher granularity of the transformations on the flows that lead to the overall achievement of the functions. For example, the function "Generate Propulsion Torque" is achieved by a sequence of function according to the following logic:

- Regulate EE (DC): If the Driver Intent is DI2, then Control Signal I8 is generated, which combines with the Stored EE (V2, I2) to supply EE (V6, I6);
- Convert EE into Torque: EE (V6, I6) is transformed to Torque (T1);
- Transmit Torque: Torque T1 is transmitted to the external actor Rear Axle.

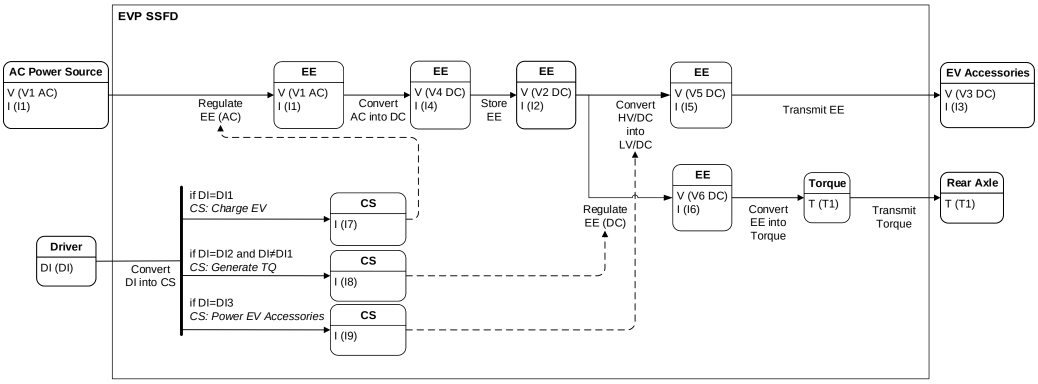


Figure 7. Nested function decomposition analysis

3.3. Integration of Nested Function Decomposition Analysis with System Design Architecture

The SSFD analysis for the EVP system has been carried out so far in the functional domain without any consideration of the system architecture design, i.e. there has been no discussion of design solutions and allocation of functions to structure. Within a systems engineering context design architecture is considered at each level of the system. For example, Figure 8 illustrates a black-box system model for the EVP, showing the inputs and outputs as states described by properties, extracted from the use case function modelling stage in Figure 5. The function modelling of the system - both use case based (Figure 5), and the integrated EVP function model in Figure 6, can be considered to be nested within the Black Box structure of the EVP system. Therefore, the EVP function model in Figure 6 can be considered a grey-box model of the EVP system, which is represented as a black-box in Figure 8.

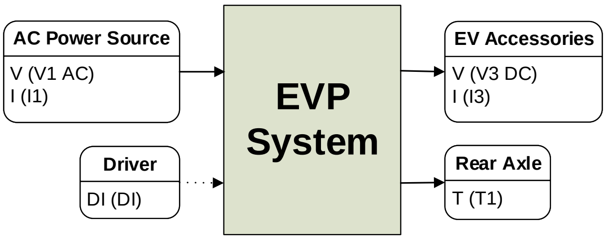
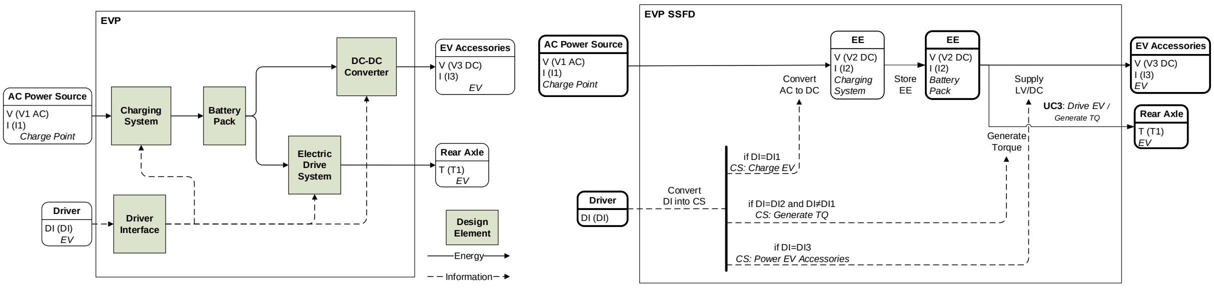


Figure 8. EVP System Black Box

Once the EVP system model analysis is complete at a given level of resolution (e.g. as illustrated in Figure 6 or 7), system design analysis can be carried out. At this level design architecture analysis relates to technology choices that are available for the system. Figure 9.a illustrates a structural design boundary diagram for the EVP, which identifies design solutions that deliver the system functions. This can be considered a white-box model of the EVP system analysed as a grey-box model in Figure 6.

Note that at this stage the design solutions are still generic, although they are limited by technology choices that have be made; e.g. the charging system will connect to a fixed plug-in charge point - as indicated in the revised input state which depicts location "charge point" (as opposed to wireless street charging while driving). The way in which the physical connection is made to the charging point has not been designed yet, but given that there will be a physical connection a logical constraint needs to be applied to prevent the drive mode while charging. This is shown in the updated EVP SSFD in Figure 9.b. Note that once the architectural design decisions are made, the linkage between functions and structures can be indicated in the SSFD by adding the "location" parameter to the definition of the state. This is illustrated in Figure 9.b in relation to the states associated with the EV charge mode.



a) EVP System Design Boundary Diagram b) Updated EVP Systems SSFD Model

Figure 9. EVP System Design Architecture Analysis

Figure 10 illustrates an updated EVP SSFD function decomposition (or subsystem level) analysis showing the nesting structure of the function chains with the design subsystems of the EVP. All states in this SSFD have now been updated with the location parameter denoting the architectural design structure of subsystems (conceptualised as physical objects) to which the functions (shown as state transitions) have been allocated.

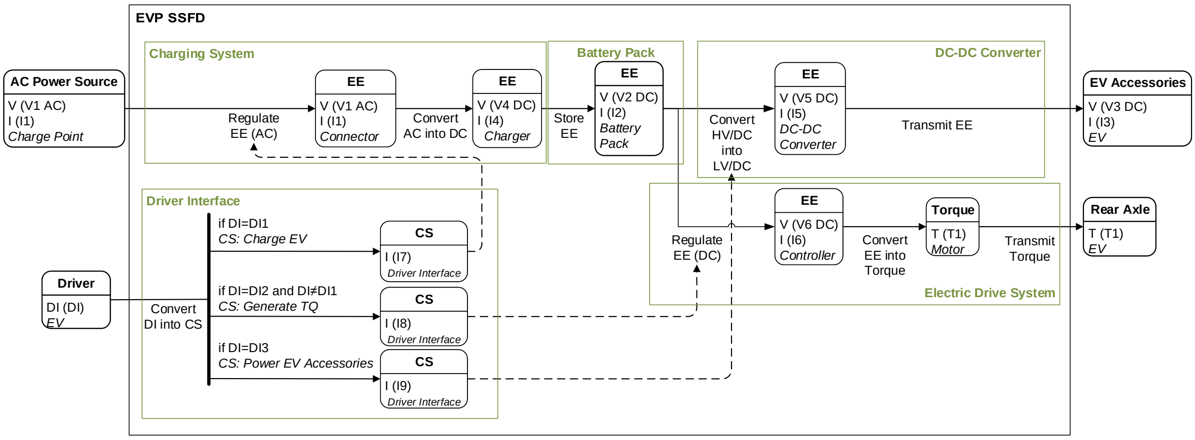


Figure 10.EVP System - Nested Function Decomposition with Design Solutions

4. Discussion and Conclusions

The increasing multi-disciplinary complexity of engineered systems and in particular the ubiquitous addition of cyber-physical features to legacy system architectures prompts for significant development of systems modelling methodologies. The contribution made by this paper is the development of a methodology for function modelling of complex systems with multiple operating modes and use cases, in a top-down, solution neutral manner. This addresses a methodological gap in function modelling practice of complex systems, with conventional approaches tending to derive design architectures for the system based on the requirements captured from the analysis of multiple use cases, without developing a solution neutral functional model. The interest for methodologies for conceptual and architecture analysis of complex multi-modal systems is well reflected by recent literature (see for example Liu et al 2013, and 2015; Mokhtarian et al, 2017).

Crilly's (2013, 2015) model of the nesting structure of systems within super-systems and the role of function modelling to map the propagation of flows through nested systems, provided a reference framework for the research presented in this paper. Our analysis centred on the function modelling aspects, and identified two distinct tasks:

- Composition through synthesis of a system functional model from multiple functional models associated with operating modes / use cases, nested within the scope of the system;
- Decomposition of the system functional model across multiple levels of abstraction, through the nesting structure that defines the architecture of the system, in an iterative top-down, solution neutral manner.

The approach presented in this paper was centred on extending and verifying the capability of the System State Flow Diagram solution neutral function modelling framework to support the integrated modelling of nested complex systems architectures, covering both tasks - i.e. model composition and decomposition - across multiple levels of abstraction.

Within a systems engineering context, the analysis starts with the identification of stakeholders and relevant use cases involving the system-to-be-designed (as system of interest). This will prompt the analyst to consider the relevant use cases and main goals associated with each use case. In the case study considered in the paper, function modelling was conducted using the SSFD methodology by modelling each use case in relation to the transformative functions that need to be achieved in order to deliver the goals and functionality required. This provided a structured and robust approach to requirements capture - as illustrated in Figure 5 for the EVP system analysis. This complements the approach commonly seen in practice as used by systems engineers, where requirements are captured based on operations view analysis of use cases, using tools such as activity diagrams, sequence diagrams and textual use case modelling. The strength of the SSFD is that it supports the analyst in clearly identify the main flows associated with each use case by defining first the input and output states through concise definition and specification of associated attributes or parameters. This environmental / stakeholder centric view of the function identifies the transformations that need to be achieved by the system, and the way of achievement is explained by the main flow decomposition on the basis of observable states in the device centric view. The application of the SSFD heuristics facilitates the identification of flows related to the function - both the main flow and auxiliary flows, and the relationships between flows, hence facilitating the development of an overall model. For example, in the EVP case study (Figure 5), the energy (EE) was identified as the main flow for each of the use cases; in addition, the SSFD also captured the way the user will interact with the system as control flows, which are linked to the main flow.

The fundamental idea behind the synthesis of functional models from multiple use cases is to identify the states that provide the coupling for the individual function models, and to aggregate the individual function models into a single overall system model, as illustrated in Figure 6. The case study provided two examples of coupling: (i) in relation to coupling of an energy flow - associated with the use of a common energy source for two completely different functions as user goals; and (ii) a logical condition coupling, based on the choice of use case / function by the driver (i.e. Driver Intent). This is important as it validates the applicability of the methodology to both physical (handling energy and material flows) and logical systems (e.g. control systems). A further remark is that the SSFD state definition which includes clear reference to state parameters, facilitates checking of the coupling conditions between the individual function models; this is important for both physical systems and logical systems.

The paper also argued and illustrated that the SSFD facilitates the function decomposition to increasing levels of granularity - before design architecture solutions are considered - through the identification of a function chain or sequence that replaces a function in a higher level function model. In terms of the SSFD representation, this simply means that we identify a number of intermediate states on the flow, which provides the increased granularity of the function model. Top-down function (or event) decomposition is a difficulty widely recognised in the systems modelling community, particularly in association with formal modelling (model specification, checking, testing) using state machine type models for cyber-physical systems (see for example Huang et al, 2016).

Figure 11 summarises the SSFD based methodology for complex system function modelling presented in this paper, with illustrations from the of the EVP case study. The layers of analysis in Figure 11 depict the phases of the function modelling illustrated for the EVP case study: (i) function analysis of individual operating modes / use cases; (ii) composition by synthesis of a system functional model from individual use case models; and (iii) system architecture analysis, driven by function decomposition and allocation of functions to subsystems / design structures. This illustrates the function driven approach to system architecting, based on the methodology presented in this paper.

Figure 11 also summarizes the different types of function modelling nesting that have been identified and addressed in this research:

- Function modelling nested with the operating modes / use cases of the system-of-interest; for the EVP this was illustrated in Figures 4 and 5 and the top layer of Figure 11;
- Nested function decomposition - illustrated by the increased granularity of function modelling for the SSFD in Figure 6, shown in Figure 7 and the middle and bottom layer of Figure 11;
- Function modelling nested with the system architecture / structures, illustrated in Figures 9 and 10, and the bottom layer of Figure 11.

The electric vehicle powertrain case study used in this paper to demonstrate the methodology was chosen because it had sufficient complexity to illustrate and prove the validity of the approach across different types of flows (energy and information) and engineering disciplines (mechanical, electrical, control / software). The methodology is now available for researchers and industry practitioners to apply for the analysis of other complex systems and in particular for cyber-physical systems where there is a recognised need for solution neutral modelling to integrate the physical and cyber domain analysis.

Acknowledgment

We acknowledge the partial support for the research presented in this paper from the "Modelling and Analysis of Cyber-Physical Systems" research project PN-III-P4-ID-PCE-2016-0210, funded by CNCS-UEFISCDI - the Romanian National Authority for Scientific Research.

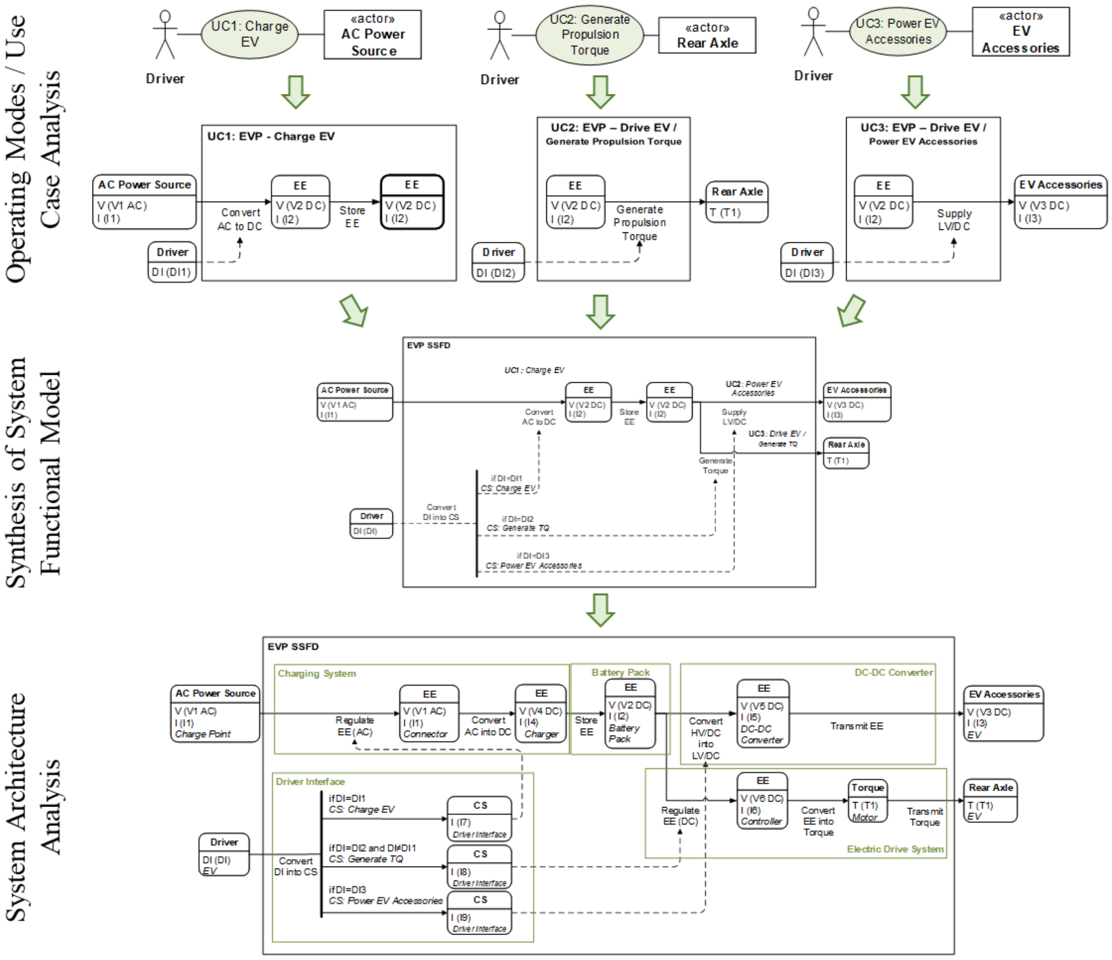


Figure 11.Nested Function Analysis and Decomposition Methodology

References

Albers, A., Wintergerst, E. (2014), "The contact and channel approach (C&C2-A): relating a system's physical structure to its functionality", In: Chakrabarti, A., and Blessing, L.T.M. (Eds.), An Anthology of Theories and Models of Design, Springer-Verlag London, UK, pp. 151-171, DOI: 10.1007/978-1-4471-6338-1_8.

Campean, I.F., Henshall, E., Brunson, D., Day, A., McLellan, R., and Hartley, J. (2011), "A structured approach for function analysis of complex automotive systems", SAE International Journal of Materials and Manufacturing, Vol. 4 No. 1, pp. 1255-1267, DOI: 10.4271/2011-01-1268.

Campean, I.F., Henshall, E., and Rutter, B. (2013), "Systems engineering excellence through design: an integrated approach based on failure mode avoidance", SAE International Journal of Materials and Manufacturing, Vol. 6 No. 1, pp. 389-401, DOI: 10.4271/2013-01-0595.

Crilly, N. (2013), "Function propagation through nested systems", Design Studies, Vol. 34 No. 2, pp. 216-242, DOI: 10.1016/j.destud.2012.10.003

Crilly, N. (2015), "The Proliferation of Functions: Multiple systems playing multiple roles in multiple supersystems", in AI-EDAM 29:01:83-92, https://doi.org/10.1017/S0890060414000626.

Dori, D. (2016), Model-Based Systems Engineering with OPM and SysML, Springer, Berlin, https://doi.org/10.1007/978-1-4939-3295-5.

Eigner, M., Dickopf, T. and Huwig, C. (2016), "An Interdisciplinary Model-based Design Approach for Developing Cybertronic Systems", Proceedings of the International Design Conference (DESIGN), Dubrovnik, Croatia, pp. 1647-1656.

Eisenbart, B., Gericke, K., Lucienne, T.M., Blessing, L.T.M., and Timothy, C.M. (2017), "A DSM-based framework for integrated function modelling: concept, application and evaluation", Research in Engineering Design, Vol. 28 No. 1, pp. 25-51, DOI: 10.1007/s00163-016-0228-1.

Friedenthal, S., Moore, A. and Steiner, R. A practical guide to SysML: The Systems Modeling Language. Burlington:Morgan Kaufmann; 2008.

Hehenberger, P., Vogel-Heuser, B., Bradley, D., Eynard, B., Tomiyama, T., and Achiche, S. (2016), "Design, modelling, simulation and integration of cyber physical systems: methods and applications", Computers in Industry, Vol. 82, pp. 273-289, https://doi.org/10.1016/j.compind.2016.05.006.

Hoang T.S., Snook C., Ladenberger L., Butler M. (2016) Validating the Requirements and Design of a Hemodialysis Machine Using iUML-B, BMotion Studio, and Co-Simulation. In: Butler M., Schewe KD., Mashkoor A., Biro M. (eds) Abstract State Machines, Alloy, B, TLA, VDM, and Z. ABZ 2016. Lecture Notes in Computer Science, vol 9675. Springer, DOI:10.1007/978-3-319-33600-8_31.

Kitamura, Y., and Mizoguchi, R. (2003), "Ontology-based description of functional design knowledge and its use in a functional way server", Expert Systems with Applications, Vol. 24 No.2, pp. 153-166, DOI: 10.1016/S0957-4174(02)00138-0.

Kossiakoff, A., Sweet, W.N., Seymour, S.J. and Biemer, S.M. (2011), Systems engineering principles and practice., 2nd edn., John Wiley&Sons, New Jersey, https://doi.org/10.1002/9781118001028.

Lindemann, U., Maurer, M., and Braun, T. (2009), Structural Complexity Management: An Approach for the Field of Product Design, Springer Verlag, Berlin, https://doi.org/10.1007/978-3-540-87889-6.

Liu, C., Hildre, H.P., Zhang, H., and Rolvag, T (2015), "Conceptual design of multi-modal products", Research in Engineering Design, Vol. 26, pp. 219-234, DOI: 10.1007/s00163-015-0193-0.

Liu, C., Hildre, H.P., Zhang, H., and Rolvag, T (2016), "Product architecture design of multi-modal products", Research in Engineering Design, Vol. 27, pp. 331-346, DOI: 10.1007/s00163-016-0221-8.

Maier, M.W., & Rechtin, E. (2000). The Art of Systems Architecting, 2nd ed. Boca Raton, FL: CRC Press

Matthiesen, S. and Ruckpaul, A. (2012), "New Insights on the Contact&Channel-Approach - Modelling of Systems with Several Logical States", Proceedings of the International Design Conference (DESIGN), Dubrovnik, Croatia, pp. 1019-1028.

Mokhtarian, H., Coatanea, E., & Paris, H. (2017). Function modeling combined with physics-based reasoning for assessing design options and supporting innovative ideation. Artificial Intelligence for Engineering, Design Analysis and Manufacturing 31(4), 476–500, DOI: 10.1017/S0890060417000403

Osorio, C.A., Dori, D. and Sussman, J. (2011), "COIM: An object-process based method for analyzing architectures of complex, interconnected, large-scale socio-technical systems", Systems Engineering, Vol. 14 No. 4, pp. 364-382, DOI: 10.1002/sys.20185.

Otto, K. and Wood, K. (2001), Product design: techniques in reverse engineering and new product development, Prentice Hall, New Jersey.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.H. (2007), Engineering Design: A Systematic Approach, 3rd edn., Springer, London, https://doi.org/10.1007/978-1-84628-319-2.

Sickafus, E.N. (1997), Unified Structured Inventive Thinking: How to Invent, Ntelleck, Michigan.

Stone, R.B., & Wood, K.L. (2000), "Development of a Functional Basis for design" Journal of Mechanical Design, Vol. 122 No. 4, pp. 359-370, DOI: 10.1115/1.1289637.

Toepfer, F. and Naumann, T. (2016), “Management of Vehicle Architecture Parameters”, Proceedings of the International Design Conference (DESIGN), Dubrovnik, Croatia, pp. 1679-1688.

Tomiyama, T., D’Amelio, V., Urbanic, J., and ElMaraghy, W. (2007), "Complexity of multi-disciplinary design", Annals of the CIRP, Vol. 56 No. 1, pp. 185-188, https://doi.org/10.1016/j.cirp.2007.05.044.

Ullman, D.G. (2010), The mechanical design process, 4th edn., McGraw-Hill, New York.

Umeda, Y. and Tomiyama, T. (1995), "FBS modeling: modeling scheme of function for conceptual design", Proceedings of Working Papers of the 9th Int. Workshop on Qualitative Reasoning About Physical Systems, Amsterdam, pp. 271-278.

Ulrich, K.T. and Eppinger, S.D. (2003), Product design and development, 3rd edn., McGraw-Hill/Irwin, NY.

van Beek, T.J., & Tomiyama, T. (2009), "Integrating conventional system views with function-behaviour-state modelling", Proceedings of CIRP Design Conference, Cranfield University, UK, March 30-31.

Weilkiens, T. (2008), Systems Engineering with SysML/UML: Modelling, Analysis, Design. Burlington: Morgan Kaufmann.

Yildirim, U., Campean, F., and Williams, H. (2017), "Function modeling using the system state flow diagram", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 31 No. 4, pp. 413-435, https://doi.org/10.1017/S0890060417000294.