FUNCTION MODELLING OF COMPLEX MULTIDISCIPLINARY SYSTEMS

Development of a System State Flow Diagram Methodology for Function Decomposition of Complex Multidisciplinary Systems

Unal YILDIRIM

Submitted for the Degree of

Doctor of Philosophy

Faculty of Engineering and Informatics

University of Bradford

2015

Abstract

Unal Yildirim

Function Modelling of Complex Multidisciplinary Systems

Development of a System State Flow Diagram Methodology for Function Decomposition of Complex Multidisciplinary Systems

Keywords: Design Methodology, Product Development, Systems Engineering, Function, Function Chain, State, State Transition

The complexity of technical systems has increased significantly in order to address evolving customer needs and environmental concerns. From a product development process viewpoint, the pervasive nature of multi-disciplinary systems (i.e. mechanical, electrical, electronic, control, software) has brought some important integration challenges to overcome conventional disciplinary boundaries imposed by discipline specific approaches. This research focuses on functional reasoning, aiming to develop a structured framework based on the System State Flow Diagram (SSFD) for function modelling of complex multidisciplinary systems on a practical and straightforward basis.

The framework is developed at two stages.

1) The development of a prototype for the SSFD framework. The proposed SSFD framework are tested and validated through application to selected desktop case studies.

2) Further development and extension of the SSFD framework for the analysis of complex multidisciplinary systems with multiple operation modes and functional requirements. The developed framework is validated on real world case studies collaborated with industrial partners.

The main conclusion of this research is that the SSFD framework offers a rigorous and coherent function modelling methodology for the analysis of complex multidisciplinary systems. Further advantages of the SSFD framework is that 1) the effectiveness of the Failure Mode Avoidance (FMA) process can be enhanced by integrating the SSFD framework with relevant tools of the FMA process, and 2) the integration of the SSFD with the SysML systems engineering diagrams is doable, which can promote the take-up of the approach in industry.

Acknowledgement

I am extremely grateful to Prof. Felician Campean, Director of the Automotive Research Centre, for both supervising and funding this research. Over the past four years he has been tremendously supportive and patient through this research project.

I would like to thank all the members of the Automotive Research Centre for all their support during my time spent as a research student, particularly fellow researcher Amad Uddin and Dr Mohammed Reza Kianifar.

Finally, I would like to thank my dad, Huseyin, and my brother, Sinan, for their support over the last four years and throughout my education.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AC	Alternating Current
ARS	Active Rear Spoiler
BEQIC	Bradford Engineering Quality Improvement Centre
C	(Electric) Current
C&C^2-A	Contact and Channel Approach
C&CM	Contact and Channel Model
CS	Control Signal
CSS	Channel and Support Structure
D	Drag Count
DC	Direct Current
D&C	Divide-and-Conquer
DFSS	Design for Six Sigma
DI	Driver Input
DMM	Domain Mapping Matrix
DOF	Degree of Freedom
DSM	Design Structure Matrix
E	Energy
EE	Electrical Energy
ESD	Enhanced Sequence Diagram
EV	Electric Vehicle

EVP	Electric Vehicle Powertrain
FAST	Function Analysis System Technique
FB	Functional Basis
FBS	Function-Behaviour-State
FBSt	Function-Behaviour-Structure
FDS	Fuel Delivery System
FFBD	Functional Flow Block Diagram
FMA	Failure Mode Avoidance
FVSC	Front View Split Camera
HV	High Voltage
IDEF	Integration Definition for Function Modelling
IFM	Integrated Function Modelling
INCOSE	International Council on Systems Engineering
I	Intent
IS	Image Status
IT	Iteration
L	Litre
LP	Loop
M	Mass
MDM	Multiple Domain Matrix
ME	Mechanical Energy
MSC	Message Sequence Charts

OAF	Object-Attribute-Function
OPD	Object-Process Diagram
OPL	Object-Process Language
OPM	Object-Process Methodology
P	Pressure
PN	Petri nets
PR	Perceived Result
PSS	Product Service System
R	Result
RE	Radiant Energy
RP	Repetition
S	Size
SBF	Structure-Behaviour-Function
SDL	Specification and Description Language
SoC	State of Charge
SS	Spoiler Status
SSFD	System State Flow Diagram
STD	State-transition Diagram
SUV	Sports Utility Vehicle
SysCARS	System Core Analyses for Robustness and Safety
SysML	System Modelling Language
T	Temperature

TQ	Torque
ThR	Thermal Radiation
TRIZ	The Theory of Inventive Problem Solving (English name)
UML	Unified Modelling Language
USIT	Unified Structured Inventive Thinking
v	Air Velocity
V	Voltage
VV	Vehicle Velocity
WS	Working Surface
WSP	Working Surface Pair
XOR	Exclusive OR

1. Introduction

1.1 Background

The complexity of technical systems has increased rapidly driven by demands for new technologies, sustainable, cost-effective and quality products (Lu and Suh, 2009). Furthermore, these demands have resulted in the development of multi-disciplinary products which have made the product development process more challenging. For example, modern automotive systems nowadays include various elements crossing different disciplines, like electro-mechanical structure with electronic, mechanical and control features embedded within the system.

The multi-disciplinary nature of technical systems presents inter-disciplinary problems in the product development (Tomiyama et al., 2007). It is difficult to address these problems, since conventional engineering education and the methods for design analysis and synthesis are discipline-wise (D`Amelio and Tomiyama, 2007). Within a customer focused engineering approach, systems engineering design must focus on robust and reliable delivery of system functional requirements. If functions are not identified, they will not be specified and engineered in, which will likely result in failures in the system (Campean and Henshall, 2012a). Complex systems can have multiple operation modes, which each have different functional requirements pertain to various disciplines, e.g. fuel engine and electric motor are used as switchable technologies in a hybrid electric vehicle. Each technology addresses different functional requirements (Liu et al., 2015).

Structure-based system decomposition is general practice in industry (Eisenbart, 2014). The way of decomposing a system based on its structure may not be effective enough due to the increased complexity and multi-disciplinary structure of systems. This shows the importance of understanding the integration of systems from different disciplines in terms of their functional structures.

There are a variety of function-based approaches to system decomposition in literature (see Erden et al., 2008). Liu et al. (2015) argue that the traditional methodology in engineering design is based on the design of systems with fixed configurations. Most of the function modelling approaches focus on the representation of one mode of operation of the systems in the context of the analysis of the overall system function. A systematic method to involve and enable the functional representation of multiple modes of operation is lacking.

1.2 Motivation

The initial motivation of this work was based on the experience with function analysis tools within an industrial engineering design environment in the automotive industry, through collaborative work undertaken over a period of more than 10 years within the Bradford Engineering Quality Improvement Centre (BEQIC). This experience, as reflected in Campean et al. (2010), was mainly focused on the robustness and reliability aspects of the design of automotive systems, within a Failure Mode Avoidance (FMA) context, which has been introduced as a structured approach to deal with failure modes early in the design process (Saxena et al, 2015). Unlike other industrial fields, which use failure mechanisms of parts and bottom-up fault propagation through the system as the basis for design risk evaluation (e.g. Tumer and Stone, 2001), the automotive industry has adopted a model based approach based on the top-down functional decomposition of the system for failure modes and effects analysis (Ford Design Institute (FDI), 2004). The advantage of this approach is that it enables consistent focus on customer required functions and, in principle, has strong alignment with the Systems Engineering "V" model (INCOSE, 2015). However, as discussed by Campean et al (2011), the predominant approach in industry is that function analysis in the context of failure mode avoidance and risk assessment in early design is not integrated with the systems engineering requirements process. This is coherent with the guidelines for tools like Failure Modes and Effects Analysis (FMEA) (FDI, 2004) which stipulate that the first step of the analysis should be to "brainstorm" functions of the system and represent functions in a function tree. The brainstorming based Function Tree approach to function analysis of a system has the advantage that it is simple to teach and implement in an industrial environment; however, it is not robust in that:

1) The integrity of the analysis (e.g. evaluated in terms of the completeness of function requirements identification) strongly depends on the expertise of the team and the time allocated for the task;

2) The structure of the functional model strongly depends on the analyst, which can have an impact on subsequent design action for countermeasure development;

3) The function analysis is conducted on a structural basis (based on a Boundary Diagram of the system), hence it can only be conducted after the system architecture design is completed. This method is not effective for solution neutral function analysis of the system, which means that invariably there will be a difference between the function modelling approach used by the initial design analysis focused on customer and logical / functional requirements for system architecting, and the functional modelling for the physical systems design and analysis.

In relation to the last point above, it is useful to reflect that historically the design of a mature system like a vehicle was predominantly iterative, with large amount of carryover and reuse. However, with the explosion of cybertronic systems in the structure of the car and the prevalence of software based features controlling enhanced functionality of hardware components, there is a fundamental need to strengthen the function modelling framework to ensure that it facilitates the solution independent analysis of functionality to support the requirements analysis and allocation across multidisciplinary systems. It is also important that the methodology is easy to deploy in a real world product development environment and process, i.e. supports all required activities from requirements specification, to initial design analysis and synthesis integrated with failure modes analysis and countermeasures development, and verification and validation.

In order to address this challenge, the System State Flow Diagram (SSFD) (Campean & Henshall, 2008; Campean et al, 2011; Campean et al, 2013b) has been introduced as a structured approach for function mapping based on the analogy with reliability block diagrams and state transition diagrams (Birolini, 2010). The SSFD has been introduced on the basis for function analysis within an integrated FMA framework for systems engineering design (Campean and Henshall, 2012a). Figure 1.1 illustrates SSFD for the function “provide power for low voltage vehicle consumers” of an electric vehicle powertrain.

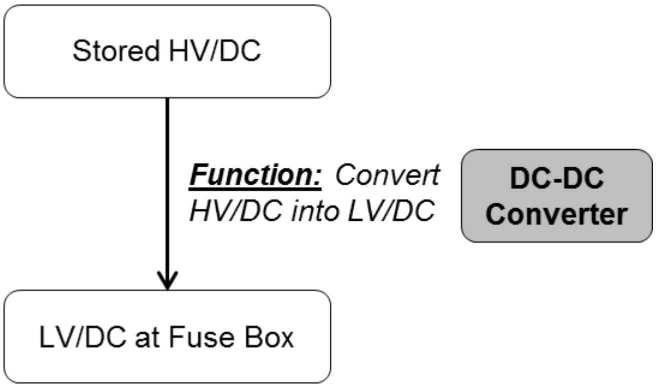


Figure 1.1: SSFD for the function “provide power for low voltage vehicle consumers” (adapted from Campean et al, 2011)

The SSFD represents a system in terms of discrete states which are described in respect of the flows of energy, material and information in the system. A state is represented by a box. Figure 1.1 shows the input and the output states as “stored high voltage (HV) direct current (DC) electric power” and “low voltage (LV) DC electric power at fuse box”, respectively. In the SSFD, the transfer between the states is provided by a function, which is articulated using a verb-noun structure and denoted by an arrow. The mapping of functional requirements onto design solutions are also represented on the diagram, as shown by a grey box in Figure 1.1 (Campean and Henshall, 2012a). The approach has been extensively taught and deployed within an industrial environment, and several case studies published argue that the approach is applicable across the engineering domains, including mechanical systems, as well as control and software systems (Campean et al., 2013b).

The SSFD offers a structured approach for system decomposition on a functional basis, and maintains the discipline of solution-independent thinking in the analysis of a system. These features of the diagram promote better design analysis and synthesis, coherent with the theoretical basis for function decomposition (Chakrabarti and Bligh, 2001) and better allocation of design responsibilities to engineering design teams on a functional basis. The SSFD also support the development of other FMA process tools (i.e. boundary diagram, function tree and interface analysis) which are commonly used in industry (Campean et al., 2013a).

While the SSFD has demonstrated strong potential for supporting function modelling within a multidisciplinary environment, a critical analysis of the methodology and its effectiveness based on the existing case studies pointed out some key limitations, summarized as follows:

1) The SSFD lacks a rigorous definition of the key elements of “state” and “function”.

2) It is mainly focused on the analysis of a main flow through the system, and case study applications so far have been at subsystems/physical systems level, rather than high level analysis of the Systems Engineering “V” where the focus is on customer required functionality (utility to the customer).

3) Complex systems, like a vehicle powertrain, commonly have multiple “main” function flows through the system, which are related with multiple operating modes of the system. While the SSFD supports analysis of multiple flows and operating modes (e.g. Campean et al. 2013a have illustrated this with an exhaust aftertreatment system covering both the normal operation and the regeneration mode), there is no rigorous definition or structured guidelines of how SSFD should be conducted in the context of complex multidisciplinary systems.

4) There is no rigorous definition on how the SSFD can be developed and deployed across systems levels. While consistent discussion based on significant case studies (Henshall et al, 2014 and 2015) of how the function analysis underpinned by SSFD is deployed successively from system to subsystem and component, there is no analysis provided on how successive levels of analysis can be integrated within a SSFD to offer a coherent and comprehensive model of the system (i.e. how a subsystem SSFD should be integrated within the higher level system SSFD).

5) The SSFD is not integrated with other systems engineering tools for requirements capture and management, in particular SysML based diagrams such as context diagrams, state machines and sequence diagrams (Friedenthal et al., 2012).

6) The SSFD has not been consistently benchmarked against other functional modelling graphical frameworks available in literature for either theoretical or empirical performance.

This analysis of the current practice of function analysis and limitations of the SSFD for the function modelling of complex multidisciplinary systems has provided the initial motivation and scope for the research. Therefore, the aim of this research was to develop a rigorous and coherent function modelling framework based on the SSFD for the analysis of complex multidisciplinary systems.

1.3 Research objectives

The following objectives have been set to address the research aim:

a) To carry out a critical review of the published academic literature on function-based approaches to system decomposition;

b) To develop a prototype for the SSFD framework for conducting function modelling of a system, with a rigorous underpinning of key concepts and elements of the framework, thus providing a sound case for the theoretical validity of the framework;

c) To test and validate the proposed SSFD framework through application to selected desktop case studies;

d) To further develop and extend the SSFD framework for the analysis of complex multidisciplinary systems, addressing the limitations discussed earlier – in particular the representation multiple flows, consistent integration of multiple levels of analysis through a nested system structure, and integration with other systems engineering graphical tools.

e) To validate the developed SSFD framework through application to selected case studies of complex systems, conducted in conjunction with industrial partners, to verify that the proposed framework is consistent and coherent, and effective in its real world application to the analysis of multidisciplinary systems with multiple operation modes;

f) To review critically the experience from the theoretical analysis and from the case studies, to present an argument for the theoretical and empirical validity of the SSFD framework, and to make recommendations for further work.

1.4 Research methodology

Grix (2004) categorizes research into inductive and deductive. The former reaches conclusions from specific empirical data and generalises these conclusions, while deductive research is a theory-driven research that uses the method of proposing hypotheses and tests the acceptability of the proposed hypotheses on empirical data.

Davis (2006) introduced design development as an iterative mapping between induction and deduction which was used as a research methodology in this thesis. Therefore, this research consists of two parts, i.e. theory-driven (deductive) and test-driven (inductive). Figure 1.2 illustrates the research methodology.

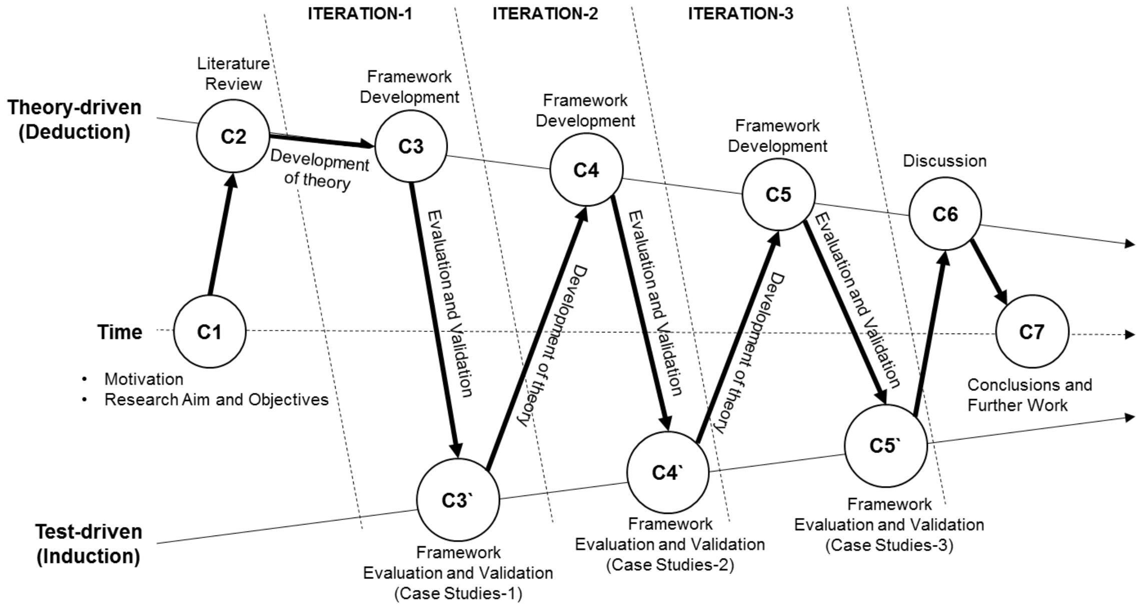


Figure 1.2: Research methodology

The first step of the methodology (C1) was to formulate research problem, aim and objectives based on the preliminary literature review and current systems engineering practice in automotive industry. An in-depth literature review (C2) was undertaken to examine the field of the problem. As shown in Figure 1.2, in respect of research objectives b-to-e in Section 1.3, three iterations were required between deductive and inductive parts of the research. These iterations can be described as follows:

- **Iteration 1:** The development of key elements of the framework based on existing literature (C3) and the validation of these elements on desktop case studies (C3`).
- **Iteration 2:** The development of key concepts of the framework for the deployment of the proposed elements to develop functional model of a system (C4) and the validation of these concepts on desktop case studies (C4`).
- **Iteration 3:** The development of a methodology for the deployment of the framework to develop functional modelling of complex multidisciplinary systems with multiple operation modes (C5) and the validation of the developed methodology on real world case studies (C5`).

There are various research methods available to test design methodologies such as the SSFD, ranging from interviews, case studies, direct observation to comparisons of data, questionnaires and statistics (Dawson, 2009). For example, Eckert et al. (2003) and Blessing and Chakrabarti (2009) emphasized the use of empirical studies which include case studies. Evaluation of the SSFD in function modelling requires multiple design projects in different companies, which is not feasible within the time available for this research project. Case-studies are one of popular research methods (Grix, 2004). Therefore, this research represents the validation of the SSFD on a range of case studies. By doing so, the usefulness of the SSFD in function modelling can be evaluated as well as its benefits and potentials can be compared to the existent function modelling approaches.

The findings from the case studies (C6) resulted in recommendations for further improvements to the SSFD (C7).

1.5 Thesis structure

Figure 1.3 shows the structure of the thesis in relation to the research methodology in Figure 1.2.

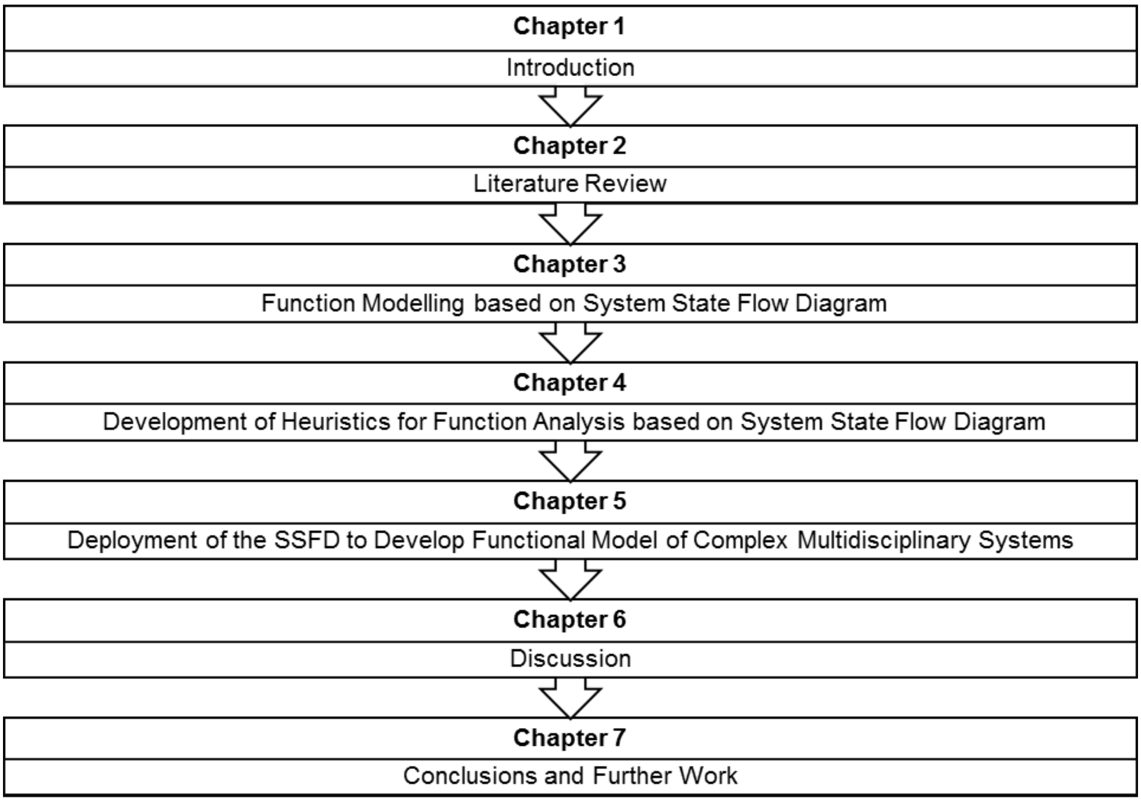


Figure 1.3: Thesis structure

As shown in Figure 1.2, Chapter 2 (C2) focuses on theory-driven part of the research and it represents a critical overview of function modelling approaches.

Chapter 3 (C3), Chapter 4 (C4) and Chapter 5 (C5) are concerned with both theory- and test-driven parts of the research. C3`, C4` and C5` in Figure 1.2 denote test-driven parts of these chapters, respectively. Hypotheses are proposed and tested in the same chapter. Figure 1.2 shows that these chapters constitute framework development and validation parts of the research and they highlight “iterative” characteristic of the methodology, i.e. the output from test-driven part (e.g. C3`) provides the input of theory-driven part (e.g. C4), and vice versa.

Chapter 3 and Chapter 4 focus on the development of a prototype for the SSFD framework. The main elements of the SSFD are represented in Chapter 3 and these elements are validated in the same chapter (C3`). Chapter 4 describes a set of steps for the development of functional model of a system using the elements of the SSFD. These steps are tested and validated on desktop case studies (C4`).

The test of the SSFD framework on a variety of desktop case studies in Chapter 4 (C4`) promotes the development of a methodology for the deployment of the SSFD framework to develop functional models of complex multidisciplinary systems with multiple operation modes in Chapter 5. This methodology is illustrated and validated on a range of case studies in conjunction with industrial partners (C5`).

As shown in Figure 1.2, Chapter 6 (C6) is theory-driven and the key developments introduced in this thesis are critically reviewed in this chapter. The research is concluded by Chapter 7 (C7) which highlights the research contributions, the main conclusions of the research and recommendations for future work.

2. Literature Review

This chapter introduces the literature research undertaken for this thesis. The chapter firstly describes the need for function modelling. This follows an overview of function modelling approaches which have the greatest relevance to the aim of the thesis. Next, the key findings are summarized in the context of common characteristics of the reviewed function modelling approaches. The chapter concludes with a critique of the reviewed approaches with the aim of clarifying the gap in the research.

2.1 Function modelling in design

Mital et al. (2008) describe design as the act of formalizing an idea into tangible information. Engineering design formalizes this idea based on customer required functions (Eggert, 2005). Wright (1998) uses the term “product design” for the definition of a product. Roozemburg and Eekels (1995) describe product design process as the documentation of the geometry, materials and production techniques of a new product, while Cross (2000) describe this process as a set of activities from the product planning to description of the refined product definition. In short, product design can simply be described as information gathering process in respect of the design of a product. Figure 2.1 shows a generic product design process of Pahl et al. (2007).

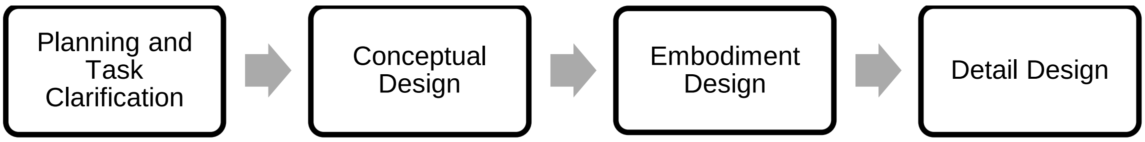


Figure 2.1: Product design process (based on Pahl et al., 2007, p.130)

The first step is to plan and clarify the task. This step results in the development of a requirement list that includes product requirements and constraints. The conceptual design step uses this list to identify essential problems and establishes function structures in respect of these problems. This follows a search of ideal working principles for the functions and then these principles are combined into a working structure. This results in the specification of a concept. The embodiment design step starts from this concept, determines the construction structure of the design and develops it in accordance with technical and economic criteria specified in the first step. The last step, the detail design, is about the preparation of production and operating documents which include detailed drawings of the individual parts (Pahl et al., 2007).

While design process of a product concerned with information gathering about the product, further steps (i.e. manufacture and sell) are required to deliver the product to the market. The extended version of product design is generally named “product development” and it is variously articulated in literature. For example, Pugh (1991) refers to it as total design incorporating product delivery process and product development process. As well as Pugh (1991), Ulrich and Eppinger (2003), Roozenburg and Eekels (1995) and Otto and Wood, 2001) emphasize the role of the product design in the product development process. Further extension of the product development process to product disposal is referred to as product lifecycle (Yang and El-Haik, 2009) or product realization process (Eggert, 2005).

The product development process is a dynamic process since the systems have an evolutionary nature due to the correlation between the increased pace of customer requirements and the increased complexity of systems. This required product development organizations to manage the product development process more effectively, i.e. less cost and development time. Systems engineering aims to address the need for an increase in the efficiency of the product development process (Frezzini et al., 2011; Cook and Wissmann, 2007). There are different systems engineering models in literature, i.e. linear, V, spiral and waterfall models (Kossiakoff et al., 2011). The “V” diagram illustrated in Figure 2.2 is commonly used in industry (INCOSE, 2015).

V diagram in Figure 2.2 provides a view of product design process in Figure 2.1 with explicit relationships shown between the process steps (left side) and the developed and validated product (right side). As shown in Figure 2.2, the requirements that drive the next step and a plan for the verification of the current level of decomposition are created at each step on the left side of the diagram. For example, during the high-level design step, a requirement document is created to drive the detailed design step, as well as a verification plan is created to drive subsystem testing. Relevant documentation (e.g. requirements for subsystem testing) is created at each step on the right side of the diagram (Shamieh, 2012).

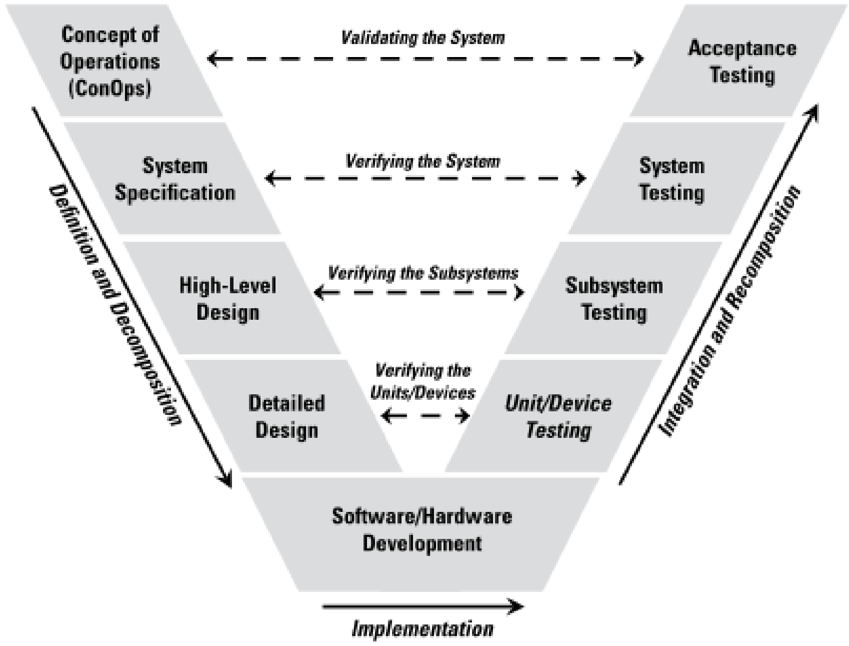


Figure 2.2: The V diagram for the systems engineering design (Shamieh, 2012, p.17)

Both product design process and systems engineering design process highlight the importance of establishing function structures of a system. Without accurate function structure to guide the process, the rest of the process is futile. System engineering design process places the emphasis on the hierarchical decomposition of functions which is described by Umeda and Tomiyama (1995) as one of the fundamental tasks in design. Coherent with the theoretical basis for function decomposition (Chakrabarti and Bligh, 2001) and the axiomatic design (Suh, 1990), the systems engineering design represents engineering design as an iterative mapping between the functional domain and the physical domain until a level of detail, where a solution concept is reached and the design can be carried out based on this concept (Campean et al., 2011).

2.2 Importance and challenges of using functional models

In design, product follows concept and concept follows function. Therefore, function structures are of great importance in the development of products that solve problems (Ullman, 2010). They underpin the search for solutions to a design problem by providing a better understanding of the problem (Pahl et al., 2007; Ullman, 2010).

The increased complexity and multi-disciplinary structure of the automotive systems, in particular the inclusion of mechatronic and control systems, have made structure-based approaches less effective in system decomposition (Campean and Henshall, 2012a). This has stressed the importance of understanding the integration of systems from different disciplines in terms of function structures. This shows that function modelling has the potential to improve communication between different disciplines in an organization (Campean et al., 2013b; Eckert, 2013).

The concept of function and the nature of function modelling approaches present challenges of using functional models. Many function modelling approaches in literature use different notions of functions and they are mostly suited to the analysis of systems from specific disciplines, e.g. mechanical systems. Function modelling approaches generally lack of clear modelling conventions and tools due to the fact that they are not well integrated with CAD system or design analysis software. Furthermore, modelling approaches are not easy to learn and easy to apply, which put off designers in industry using them (Eckert, 2013). Tomiyama et al. (2013) suggest that practitioners do not recognize very well the merit of applying function modelling to design and therefore they do not generally use function modelling in practice. Eisenbart (2014) pointed out that the implementation of in-house developed function modelling approaches in industry rely on the personal preferences of the designers, i.e. the sequence of modelling steps is not structured and the approaches are applied on the design of sub-systems or systems depending on the novelty of the design.

Numerous approaches in engineering design literature support the development of function structure of a system. The next section provides an overview of these approaches.

2.3 An overview of function modelling approaches

2.3.1 Function modelling approaches in literature

Erden et al. (2008) use the term "Function Modelling" to refer to the activity of developing models of systems based on their functionalities and the functionalities of their sub-components. Different terms are used in literature with similar meaning such as Pahl et al. (2007) use the term "function structure". The term "function modelling" used henceforth encompasses the associated terms used in the development of functional model of a system. Several researchers presented reviews of function modelling approaches from different points of view.

- King and Sivaloganathan (1998) classified existing function analysis methods and techniques into five areas of application, i.e. value analysis, failure analysis, concept analysis, artificial intelligence and function classification.
- Chiang et al. (2001) reviewed function modelling approaches with a strong focus on the way of reasoning and representing functions.
- Erden et al. (2008) presented a review of function modelling approaches by using 17 criteria which are classified under six items, i.e. Ontology, Semantic Definition of Function, Function Representation Formalism, Function-Context Relation, Decomposition and Verification, Implementation in a Programming Environment and Application.
- The review of Srinivasan et al. (2012) focused on chronological development of function definitions and function representations. They deducted four views of function, i.e. Level of Abstraction, Requirement-Solution, System-Environment and Intended-Unintended.
- Summers et al. (2013) proposed that there are three dimensions of function modelling approaches, i.e. representation characteristics, supported cognitive dimension characteristics and enabled reasoning activities. They suggest that function modelling approaches should be benchmarked based on these dimensions.
- Eisenbart (2014) provided a discipline-focused review of function modelling approaches, i.e. Mechanical Engineering, Electrical Engineering, Software Development, Service Development, Mechatronic System Development, Product-Service Systems Design and Systems Engineering.

Suh (2001) describes four domains in engineering design: customer domain, functional domain, physical domain and process domain. The design is carried out iteratively from the domain on the left (i.e. customer) to the domain on the right (i.e. process). The problem of mapping of functions in the functional domain is to provide a coherent cascade of functions of a system for the achievement of its overall function with related to the customer domain. Suh (2005) relates the complexity of a system to the measure of uncertainty in achieving its functional requirements. Most systems in real life possess a degree of complexity, as they rely on the fulfilment of multiple function chains (set of connected functions) in connection with each other. This demonstrates the importance of the development of function chains of a system in a structured way. Therefore, this chapter reviews function modelling approaches by classifying them into four main perspectives in terms of the way of developing function chains of a system. These perspectives can be useful in highlighting issues of the current approaches in function modelling and they clarify the positioning of the work presented in this thesis. These four main perspectives can be described as follows:

- **Task-oriented approaches:** They identify “what” (i.e. function) must happen without assuming “how” (i.e. structure) must happen. The functions are described and represented with respect to causality.

- **Flow-oriented approaches:** These approaches focus on “what”. The essential difference to the task-oriented approaches is the mapping of functions on the basis of the flow of material, energy and information/signal through the system.

- **Function-Structure-oriented approaches:** Some approaches address “what” and “how” concurrently.

- **Function-Behaviour-Structure-oriented approaches:** These approaches introduce the concept of "behaviour” to establish a link between “what” and “how”.

The ways of developing function model of a system in literature can be analysed in a structured way through these perspectives, while existing review papers mainly focus on the analysis of the structure of the approaches by categorising them into the field of application (King and Sivaloganathan, 1998), common characteristics, e.g. function representation (Chiang et al., 2001; Erden et al., 2008; Srinivasan et al., 2012; Summers et al., 2013) and the disciplines of engineering (Eisenbart, 2014). The perspectives of these review papers are useful to elicit information on developing function models; therefore, these perspectives constitute the basis of the analysis of function modelling approaches in this chapter in order to determine how they develop function models. The key criteria that arise from these perspectives are:

- Common characteristics
    - Function definition - How the approaches express the meaning of the function?
    - Function articulation - How the approaches represents the function textually?
    - Function representation - How the approaches represents the function graphically?
    - Function decomposition - How the approaches break the overall function into sub-functions?
- The field of application - What is the field of application of the approaches?
- The disciplines of engineering - What is the applicability of the approaches to current engineering disciplines?

Numerous function modelling approaches can be categorized into each of the main function modelling perspectives described above. It is not feasible to exhaustively review all the approaches; therefore, this chapter focuses on the key approaches that run through most of the published work in function modelling. Approaches for functional decomposition are specifically reviewed to provide a holistic view on the link between the upper and lower levels of system design and description. For example, for Weilkiens (2006), Functional Flow Block Diagram (hereafter FFBD) is a popular approach in systems engineering and its review is shown under the heading “task-oriented approaches”. Figure 2.3 summarizes the reviewed approaches in each function modelling perspective.

Figure 2.3 covers a wide range of function modelling approaches developed from 1950s (i.e. Functional Flow Block Diagram) to 2000s (i.e. Contact and Channel Approach). The approaches in each perspective are represented chronologically to exploit the developments throughout time.

Task-oriented approaches
• Functional Flow Block Diagrams (FFBD) • Function Analysis System Technique (FAST) • Design Structure Matrix (DSM) • Integrated Function Modelling framework (IFM)
Flow-oriented approaches
• Pahl and Beitz • Ullman • Ulrich and Eppinger • Functional Basis (FB) of Stone
Function-Structure-oriented approaches
• Integration Definition for Function Modelling (IDEF) • TRIZ & USIT • Statecharts • Contact and Channel Approach ($C\&C^2$-A)
Function-Behaviour-Structure-oriented approaches
• Function-Behaviour-Structure (FBSt) model • Function-Behaviour-State (FBS) model • Structure-Behaviour-Function (SBF) model • Object-Process Methodology (OPM)

Figure 2.3: Function modelling approaches reviewed in this chapter

Approaches to support a particular characteristic of function modelling (e.g. function definition) are also reviewed at the end of the chapter under the headline of the relevant characteristic (e.g. function definition). This follows the review of relevant diagrams of System Modelling Language (hereafter SysML) since the use of SysML has been extended to many function modelling approaches.

Section 2.4 gives a summary of the reviewed approaches in each main function modelling perspective on the basis of their common characteristics (see Table 2.1), while Section 2.5 presents a critique of the reviewed function modelling approaches with the aim of clarifying the gap in the research.

2.3.2 Task-oriented approaches

2.3.2.1 Functional Flow Block Diagram

Functional Flow Block Diagram (FFBD) was developed in the 1950s (Weilkiens, 2006). The diagram represents functional representation of a system in a solution-neutral way by displaying the entire network of functions of the system in a sequential relationship that leads to the achievement of its overall function (NASA, 2007). Figure 2.4 shows the top level FFBD of the entire flight mission of a spacecraft and its second level FFBD based on the function “perform mission operations.”

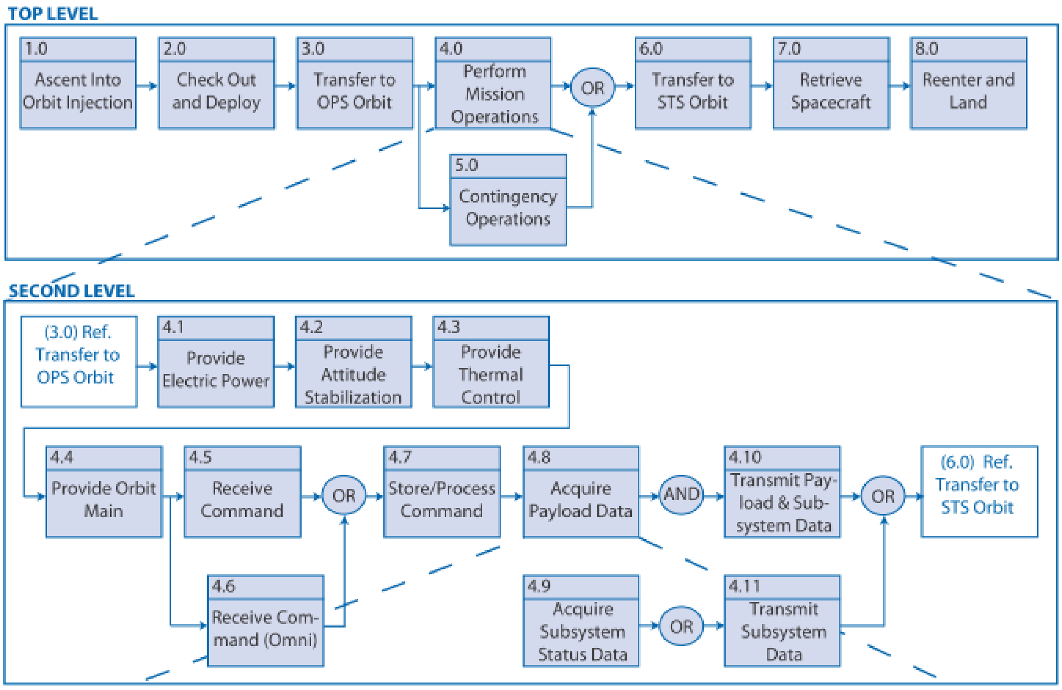


Figure 2.4: FFBDs of the entire flight mission of a spacecraft (NASA, 2007, p.53)

The FFBD describes a function as an action to be accomplished by system elements (US DoD, 2001). As shown in Figure 2.4, the diagram represents a function by a block and articulates it by an action verb (e.g. perform) followed by a noun phrase (e.g. mission operations) (FAA, 2006). Each function block is numbered according to its level such as 1.0 for the top level, 1.1 for the second level and 1.1.1 for the third level. “AND” and “OR” in Figure 2.4 are referred to as summing gates and denoted by a circle. AND is used to show parallel functions must be satisfied to proceed, while OR is used to show that alternative paths can be followed to proceed (US DoD, 2001).

The FFBD is a popular approach in systems engineering (Weilkiens, 2006) and it is supported in System Modelling Language (hereafter SysML) activity diagram, which is a graphical modelling language (Friedenthal et al., 2008). Bock (2006) details the mapping between the diagrams. While Arlitt et al. (2011) pointed out that the FFBD promotes an organization for system functions by decomposing them hierarchically, they questioned standardization of the diagram and its traceability to components. Pineda and Smith (2011) noted that time is not associated with functions of the diagram.

2.3.2.2 Function Analysis System Technique

Function Analysis System Technique (hereafter FAST) was developed by Charles Bytheway in the 1960s (King and Sivaloganathan, 1998). Like the FFBD, the FAST provides hierarchical decomposition of functions, but by applying how-why-when questions to each function. For Kaufman and Woodhead (2006), a function is an intent or causal action and it is articulated by using an active verb (e.g. support) and measurable noun (e.g. weight). Figure 2.5 shows FAST model of a mousetrap.

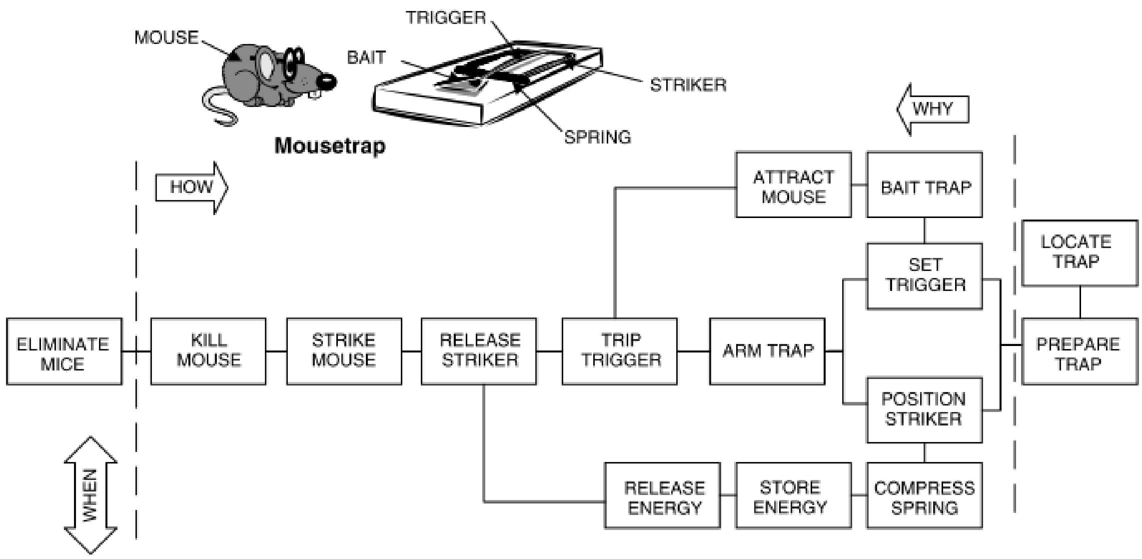


Figure 2.5: FAST model of a mousetrap (Kaufman and Woodhead, 2006, p.8)

The basic function "kill mouse" is decomposed into its sub-functions by applying how-why-when questions to the basic function and so forth. As shown in Figure 2.5, scope lines are denoted by two vertical dotted lines showing the scope of the model, that aspect of the model with which the product design and development team is interested in (Kaufman and Woodhead, 2006).

The FAST is mainly used in value engineering and therefore it focuses on the value of functions to increase the product value (Umeda et al., 1990). The FAST is also applicable in the development of new and existing products (Kaufman and Woodhead, 2006). Figure 2.5 shows solution-dependent analysis of an existing product. King and Sivaloganathan (1998) pointed out that there may be considerable difficulty in ordering the functions in a logical way in relation to the operation of the system.

2.3.2.3 Design Structure Matrix

Research on matrix-based approaches is based on the work of Steward (1981) who brought in the term “Design Structure Matrix” (hereafter DSM) to analyse the design structure of a system (Lindemann et al., 2009; Kreimeyer and Lindemann, 2011). A DSM is a square matrix displaying the elements in a system with their interactions (Eppinger and Browning, 2012). The DSM addresses interactions between the elements which belong to the same domain, e.g. functions of a product. Browning (2001) categorizes DSMs into component-based, people-based, activity-based and parameter-based. DSMs are also classified based on the assessment of interactions between system elements. A binary DSM only represents the existence of an interaction between two elements, while a numerical DSM provides a value to represent the strength of an interaction (August et al., 2005). Figure 2.6 shows a binary DSM for functions of a product.

DSM	F1	F2	F3	F4	F5
Function 1 (F1)		X			
Function 2 (F2)			X		
Function 3 (F3)				X	
Function 4 (F4)					X
Function 5 (F5)					

Figure 2.6: A binary DSM for functions of a product (Adapted from Kreimeyer and Lindemann, 2011, p.48)

Four classic techniques in the analysis of a DSM are sequencing, tearing, banding and clustering (see Kreimeyer and Lindemann (2011)). Figure 2.6 shows the technique of “sequencing”. Functions in the rows and the columns of the matrix is arranged in a way that relations between the functions are kept to a minimum as possible below the diagonal, thus reducing the complexity of the product.

The DSM is rarely used in function modelling (e.g. Chiriac et al. (2011) use DSMs in function decomposition), however the use of DSMs has been extended to many other fields of engineering design, e.g. engineering change management (Jarratt, 2004; Keller, 2007; Clarkson et. al, 2001a and 2001b), product development (Pimmler and Eppinger, 1994; Pimmler, 1994) and systems engineering (Brady, 2002). Browning (2001) emphasizes the increased use of DSMs in industrial practice including building construction, automotive, aerospace and electronics industries. The extension of DSM to Domain Mapping Matrix (hereafter DMM) (Danilovic and Browning, 2007) and Multiple Domain Matrix (hereafter MDM) (Maurer and Lindemann, 2008) has enabled matrix methodology to analyse the interactions between system elements that are from different domains. Hamraz et al. (2013) use the MDM in the representation of the function-behaviour-structure linkage method.

2.3.2.4 Integrated Function Modelling framework

Eisenbart et al. (2013a) proposed Integrated Function Modelling (hereafter IFM) framework to support cross-disciplinary function modelling of a system. The framework identified six views associated with well-known function modelling perspectives across disciplines, i.e. use case view, state view, interaction view, actor view, effect view and process flow view. These views are integrated with each other by representing them on the DSMs and the MDMs. Coherent with Vermaas (2013) and Eckert (2013), Eisenbart et al. (2014) described a central notion of function across disciplines as the intended behaviour of a system and discussed that this behaviour may be regarded through consideration of these views, for example, associated state changes of involved operands or operators in the state view.

Eisenbart et al. (2014) mentioned that the framework can be applied in different ways, for example, it can be started from different modelling activities. Eisenbart et al. (2013b) exemplified this on a range of design approaches across disciplines. According to Eisenbart (2014), potential modelling activities are carried out in the following order: use case definition, process flow modelling, operand state modelling, effect modelling, actor allocation, actor state modelling and interaction specification. Figure 2.7 represents process flow view for the use case “preparing a cup of coffee” of a customary coffee vending machine.

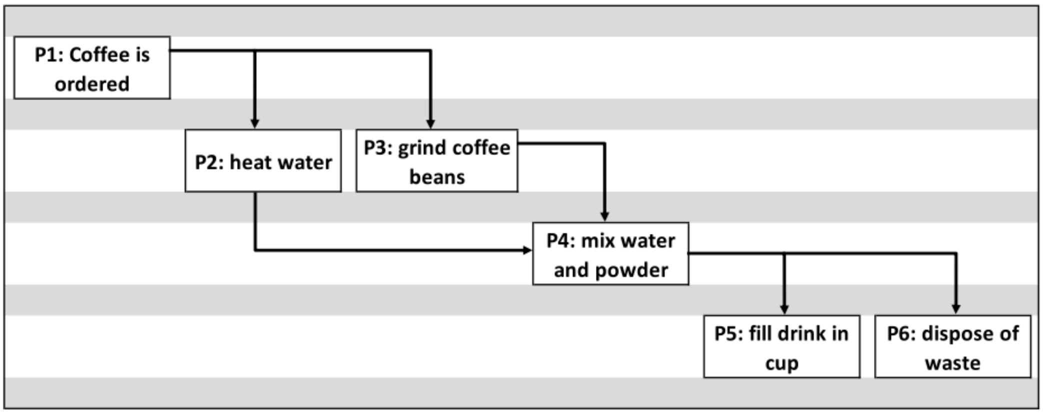


Figure 2.7: Process flow view for the use case “preparing a cup of coffee” (Eisenbart, 2014, p.125)

Figure 2.7 shows a set of sequential and parallel transformation processes for the fulfilment of the use case. Each process is denoted by a chronologically numbered block. As shown in Figure 2.7, the processes are represented related to time in the vertical direction and from left to right in the horizontal direction. The former direction is linked to the state view, while the latter is connected to the actor view and the use case view (see Eisenbart (2014) for the complete model). Eisenbart (2014) discusses the potential of the framework for supporting function analysis, e.g. conflict analysis and change prediction. Eisenbart et al. (2015) compare the IFM with SysML with the aim of improving its applicability to function modelling in interdisciplinary design.

2.3.3 Flow-oriented approaches

2.3.3.1 Pahl and Beitz

Design approach of Pahl and Beitz (Pahl and Beitz, 1988; Pahl et. al, 2007) aims to establish solution-neutral function modelling of a system. The approach starts by formulating the crux of the overall problem/requirement to be addressed by the system. Next, the overall function of the system is described based on this problem/requirement and it is broken down into sub-functions. Individual sub-functions are combined into a single functional model, implementing the overall function. Figure 2.8 shows schema of the approach.

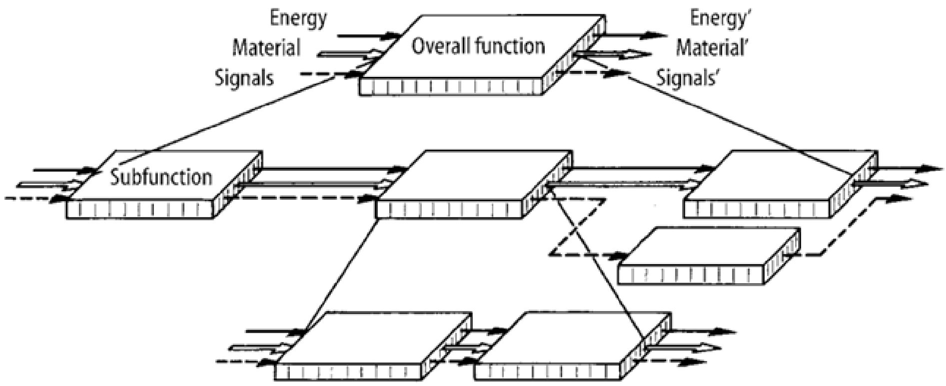


Figure 2.8: Schema of Pahl and Beitz approach (Pahl et. al, 2007, p.32)

Pahl and Beitz (Pahl et. al, 2007) describe a function as the intended input-output relationship of a system that serves to perform a task. As shown in Figure 2.8, the function is graphically represented by a block diagram (i.e. black box). A function statement is articulated in verb-noun structure and shown within the diagram. The intended inputs and outputs of the system are shown on the diagram in terms of the flows of energy, material and signal. The approach expresses solution-neutral formulation of a function by articulating it in respect of the relationship between the inputs and the outputs of the block diagram.

Pahl et. al (2007) categorize sub-functions into main functions and auxiliary functions. The main functions directly address the fulfilment of the overall function, while the auxiliary functions contribute to the overall function indirectly. The auxiliary functions are generally determined based on the nature of the solutions (i.e. design element) for the main functions. In the development of function model of a technical system, the approach starts by identifying the main flow in the system in respect of the working principle of the system. This flow includes the main functions. Once the main flow is developed, the auxiliary flows with their sub-functions (i.e. auxiliary functions) are considered. The sub-functions are represented causally through the inputs and the outputs of each function. The essential difference to the task-oriented approaches in Section 2.3.2 is the mapping of the sub-functions on the basis of the flows of energy, material and signal. Once the functional model is developed, design elements are determined for the sub-functions and they are combined into a single working structure that implements the overall function (Pahl et. al, 2007).

The method of Pahl and Beitz supports the development of new systems and existing systems. Pahl et al. (2007) suggested that the process of function decomposition can be discontinued at a high level in the case of the analysis of existing systems. If the aim is to develop a new system, the decomposition must be continued until the search for design elements seems promising. Pahl and Beitz are the driving force behind many design methods across disciplines, e.g. Ullman (2010), Ulrich and Eppinger (2003) and Stone and Wood (2000). Eisenbart (2014) points out that the basic principles of the approach have been widely adopted in mechanical engineering, electrical engineering, mechatronic system development and Product Service System (PSS) design literature.

2.3.3.2 Ullman

Ullman (1992 and 2010) focuses on mainly mechanical design process. The approach represents what the system-to-be does in terms of the flow of energy, material and information. Ullman (2010) emphasized that energy and material must be conserved as they flow through the system, that is, inputs to each function must match the outputs of the preceding function. Like Pahl et al. (2007), the overall function of the system is generated based on customer requirements and it is represented in a black box. The inputs and outputs of the system are shown on the box in terms of the flows of energy, material and information.

Ullman (2010) notes that a system may have multiple types of operating sequences while in use. Therefore, the overall function is decomposed into sub-functions by thinking of each function in terms of its preparation, use, and conclusion. The logical flow of these sub-functions addresses the fulfilment of the overall function. According to Ullman (2010), this flow can be determined by categorizing the functions into the groups of preparation, uses, and conclusion. The link between the output of one function and the input of another can be established by arranging these functions within each group. The decomposition of functions is discontinued if the sub-functions can be implemented by existing objects (i.e. design elements) or new objects are needed for further refinement. If the decomposition is complete, concepts may be generated to address all the functions. While Ullman (2010) proposes a different functional decomposition methodology from Pahl et al. (2007), concept generation methods are akin to Pahl et al. (2007), e.g. the selection of design elements for each sub-function can be made out of a set of alternatives by use of morphological method. Figure 2.9 shows functional model for a one-handed bar clamp at a high level.

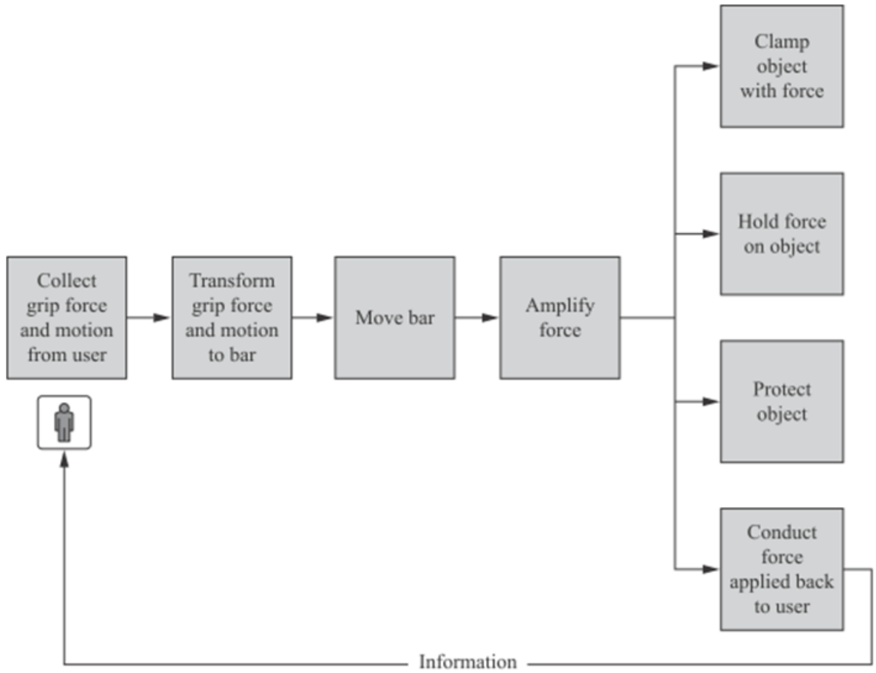


Figure 2.9: Functional model for a one-handed bar clamp (Ullman, 2010, p.187)

A function can be defined on the basis of the flow of energy, material, or information between objects or the change of an object state caused by one or more of the flows (Ullman, 2010). Figure 2.9 reflects the former. As shown in Figure 2.9, a function statement begins with an action verb (e.g. move) which is followed by a noun phrase (e.g. bar).

2.3.3.3 Ulrich and Eppinger

Ulrich and Eppinger (2003) introduce functional decomposition as a part of their concept generation method. They focus on how to use functional decomposition in the division of a problem into sub-problems. The methodology of Ulrich and Eppinger (2003) is similar to Pahl et al. (2007). The problem is formulated as the overall function of the system and it is represented as a black box with input-output flows of energy, material and signal. The overall function is decomposed into sub-functions describing what the elements of the system might do so as to fulfil the overall function. Figure 2.10 shows functional model of a hand-held nailer.

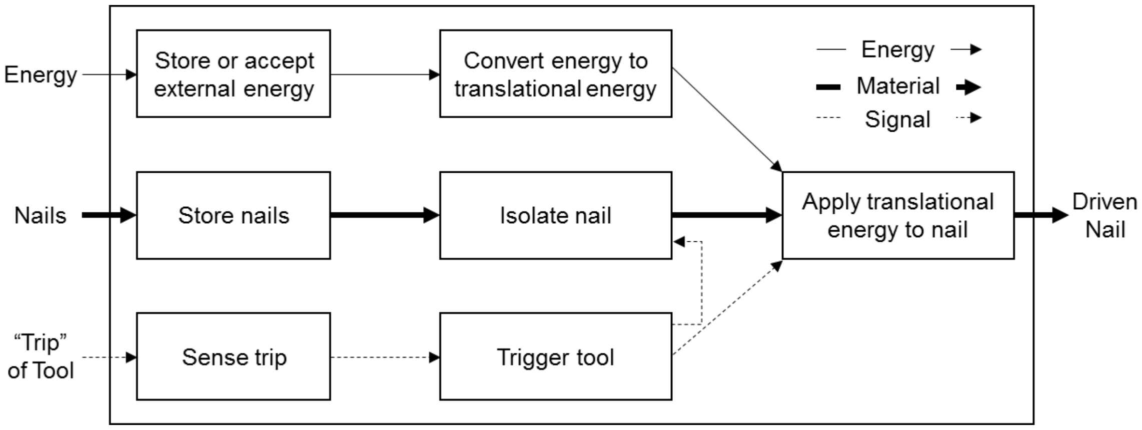


Figure 2.10: Functional model of a hand-held nailer (adapted from Ulrich and Eppinger, 2003, p.102)

The decomposition process continues until the level of sub-functions seems promising. For Ulrich and Eppinger (2003), the creation of 3-10 sub-functions at one level is ideal as a rule of thumb. The methods suggested by Ulrich and Eppinger for generating solution concepts akin to Pahl et al. (2007) and Ullman (2010), e.g. concept combination table is a type of morphological method. Furthermore, Ulrich and Eppinger (2003) introduced the concept of product architecture which arranges functions into physical chunks (see Ulrich, 1995).

Apart from functional decomposition, Ulrich and Eppinger (2003) mention that the decomposition by sequence of user actions and customer needs are alternative approaches to problem decomposition. The former involves functions in relation to user interaction, while the latter is particularly useful in the analysis of existing systems.

2.3.3.4 Functional Basis of Stone and Wood

Functional Basis (hereafter FB) of Stone and Wood (2000) builds functional models of a system based on customer requirements, like the previous flow-based approaches. However, Stone (1997) places great emphasis on the link between customer requirements and a functional model. Therefore, the approach starts by identifying flows that address customer requirements. The customer requirements are formulated in terms of the flows of energy, material and signal. Otto and Wood (2001) brought in the notion of constraint to document requirements which are not directly related to any type of flow, e.g. a customer requirement for a product to be low cost. They note that a constraint is a holistic property of a product, namely it requires consideration of the whole product.

Stone and Wood (2000) distinguish between overall function (i.e. product function) and sub-function. Once the identification of the flows for the customer requirements is complete, they are represented on a black box showing the overall function with input-output flows. The function is expressed in verb-object form. 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 on the flow through the system accordingly (Otto and Wood, 2001). Each sub-function addresses an operation on the flow of energy, material or information, which is articulated by using an active verb and an object denoting the flow. 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 2.11 shows a functional model for a common slotted bread toaster.

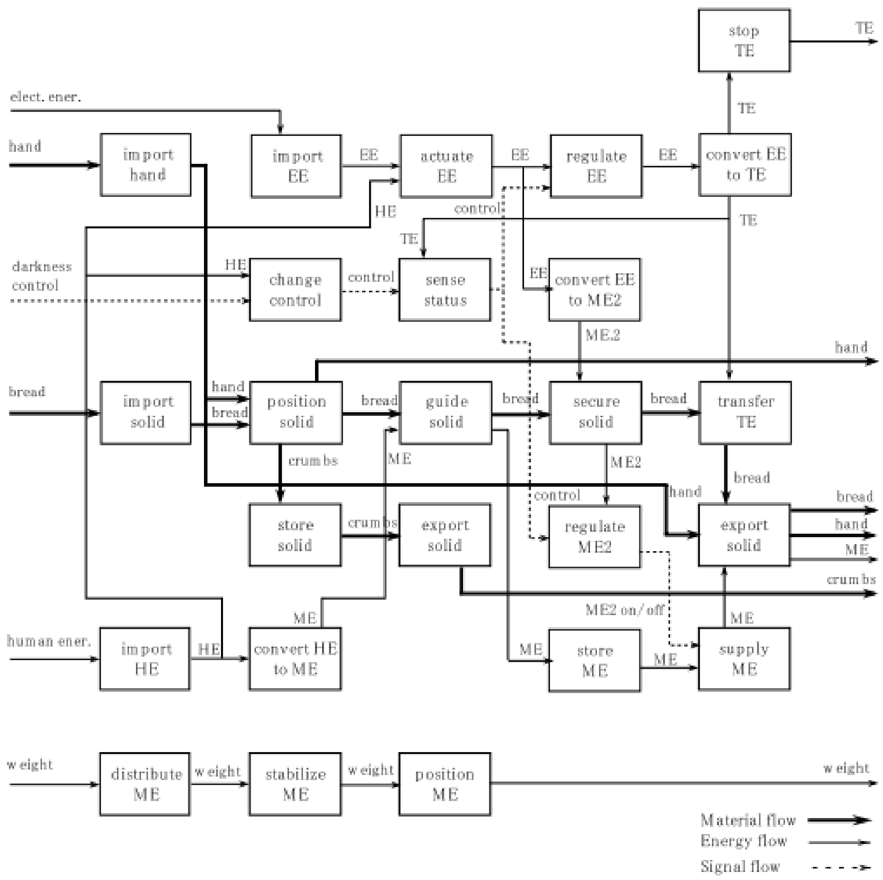


Figure 2.11: A functional model for a bread toaster (Kurfman, et al.,2000, p.9)

The FB model introduces a taxonomy of both the functions and the flows. The process of decomposing the overall function into sub-functions is continued until all sub-functions can be described by the standardized set of functions and flows (Stone, 1997).

Otto and Wood (2001) extended the concept of product architecture of Ulrich and Eppinger (2003) by introducing the notions of function dependencies and module heuristics (see also Stone et al., 2000). Function dependency is about identifying parallel and sequential function chains on a developed functional model. This facilitates the process of clustering sub-functions into modules. For Stone (1997, p.39), module heuristics are “a method of examination in which the designer uses a set of steps, empirical in nature, yet proven scientifically valid, to identify modules in a design problem.” Otto and Wood (2001) described three module heuristics which are incrementally applied to the function model to cluster the sub-functions into modules. These module heuristics reflect the flow and the function types in a functional model. The first heuristic, the dominant flow heuristic, defines “the set of sub-functions, which a flow passes through, from entry or initiation of the flow in the system to exit from the system or conversion of the flow within the system” (Otto and Wood, 2001, p.380). The branching flow heuristic identifies flows that branch into or out of parallel function chains. The last heuristic, the conversion-transmission module heuristic, addresses conversion sub-functions and conversion to transmission chains. Concepts are generated for the modules which are identified through these heuristics.

The FB mainly focuses on the mechanical and electromechanical domains (Stone and Wood, 2000). There are several publications in literature on the use of the FB in related to different fields of engineering design, e.g. risk assessment (Lough et al., 2009) and failure mode analysis of systems (Tumer and Stone, 2003; Tumer et al., 2003). Vermaas (2007) made some critical remarks on the FB with a focus on the consistency of the descriptions of the overall functions and the sub-functions. Vermaas (2008 and 2009) questioned whether the functions should obey conservation laws and always have inputs and outputs.

2.3.4 Function-Structure-oriented approaches

2.3.4.1 Integration Definition for Function Modelling

2.3.4.1.1 IDEF0

Integration Definition for Function Modelling (hereafter IDEF) is a family of modelling methods. 16 IDEF methods have been developed so far to model different types of problems, i.e. IDEF0 to IDEF14, including IDEF1X. In 1981, IDEF0 was introduced as the first method and it is widely used (O`Donovan et al., 2005).

IDEF0 is a function modelling method. A function (also called activity) is denoted by a box and articulated in verb-noun phrase. Each box is numbered in the context of the model (see Figure 2.12). A function box turns inputs into outputs. Inputs enter the function box from the left and outputs leave from the right. Controls guide the transformation process and they enter from the top, while mechanisms enter the box from the bottom and they represent design elements to achieve the function (Buede, 2009).

Function modelling of a system by the IDEF0 starts with the representation of a top-level context diagram. This diagram addresses the top-level function of the system and it is represented as a single function box, with its bounding arrows, i.e. input, output, control and mechanism (NIST, 1993). The top-level function represented on this context diagram may be decomposed into its sub-functions. The first-level decomposition of an elevator top-level function (i.e. provide elevator services), called a child diagram (NIST, 1993), is shown in Figure 2.12. This diagram preserves the bounding arrows of the top-level context diagram of the elevator (see Buede, 2009, p.62). Buede (2009) suggests that the number of sub-functions for a diagram should be limited to six or seven for its legibility.

IDEF0 has been applied to a wide range of business processes, including design and manufacturing (O`Donovan et al., 2005). Unlike the FFBD, the IDEF0 contains information relating to the flow of data between functions and physical resources that fulfil the functions (US DoD, 2001; Long, 2000). The IDEF0 provides strong hierarchical structures of a system by decomposing its activities to a lower level of detail, however King and Sivaloganathan (1998) and Eisenbart (2014) point out that the use of the method in function modelling can become complicated even in the analysis of a simple system. According to Durugbo et al. (2011), function modelling of a system by the IDEF0 can be a time-consuming activity and it is difficult to integrate the IDEF0 with related design methodologies.

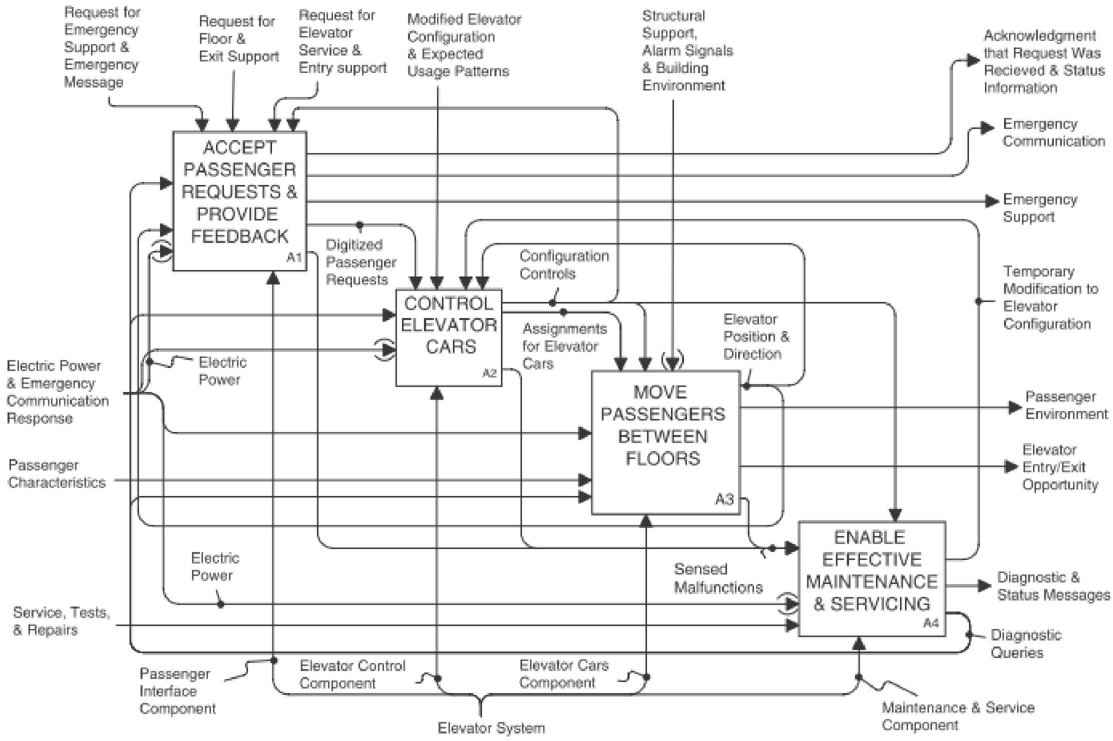


Figure 2.12: A child diagram of the elevator top-level function (Buede, 2009, p.63)

2.3.4.1.2 IDEF3

In addition to IDEF0, IDEF3 has particular relevance for function modelling. It provides two schematics regarding process and object state-transition modelling of a system (Buede, 2009). Figure 2.13 shows the process of heating water in terms of object transitions.

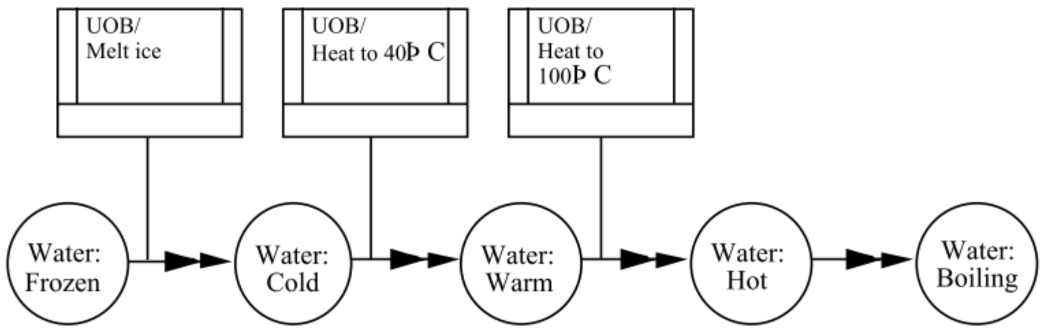


Figure 2.13: Object transitions during the process of heating water (Mayer et al., 1995, p.70)

An object is denoted by a circle containing a label, e.g. water. A corresponding state is used to represent the type or class of the object, e.g. frozen. A transition is represented by an arrow. Double headed arrow in Figure 2.13 is used to show a strong link. Units of behaviour (UoB) box represents a process which is often expressed in terms of a verb (e.g. heat) with a property (e.g. 40ºC) (Mayer et al., 1995).

2.3.4.2 TRIZ & USIT

The Theory of Inventive Problem Solving (Russian acronym: TRIZ), developed by Altshuller (1984), deals with design as an inventive problem (Nix et al., 2011). Function modelling is used in the definition of a problem. TRIZ represents function model of a system by its functional analysis diagram and substance-field functional model tools (Yang and El-Haik (2009).

2.3.4.2.1 Functional analysis diagram

A functional analysis diagram is a graphical tool that shows the flow of an action from the source of the action (Subject) to the action receiver (Object). Figure 2.13 shows a functional analysis diagram for the function “brush teeth.”

Figure 2.14: A functional analysis diagram (Yang and El-Haik, 2009, p.287)

The function is stated as an action that is denoted by an arrow and articulated in verb phrase. An action is categorized into normal useful, insufficient useful, excessive useful and harmful and it is often being achieved by applying a kind of field such as mechanical field (Yang and El-Haik (2009). In Figure 2.14, the action “brush” is achieved by toothbrush (i.e. the action source) through applying a kind of mechanical field (i.e. Mech.) on teeth (i.e. the action receiver).

2.3.4.2.2 Substance-field functional model

Substance-field functional model describes a function as an interaction between two elements of a system (Fey and Rivin, 2005) and represents the function by a triangle whose corners represent substances (S) and a field (F) (Yang and Zhang, 2000). The term “substance” is used to describe a technological system with a degree of complexity, e.g. a ship. The term “field” is referred to as a form of energy. Substances interact with each other via fields. A substance can generate a field and convert a field into another (Fey and Rivin, 2005). Figure 2.15 shows the usage of a hammer as an example.

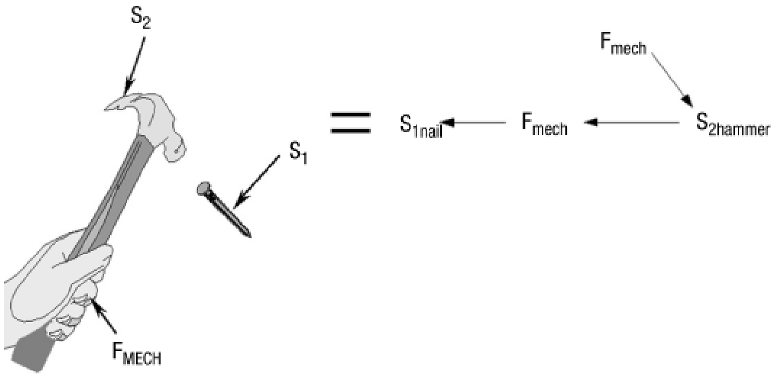


Figure 2.15: Substance-field functional model of a hammer (Fey and Rivin, 2005, p.50)

As shown in Figure 2.15, F_{mech} (mechanical energy) is applied to the hammer (S_2), which directs the energy to the nail (S_1).

The extension of TRIZ to Unified Structured Inventive Thinking (hereafter USIT) by Sickafus (1997) aimed at simplifying the process of devising solutions for engineering problems. The USIT achieves this by identifying conceptual solutions for engineering problems through converting a technological problem to a conceptual problem and then seeking all possible conceptual solutions to this problem. The object-attribute-function (hereafter OAF) framework of the USIT has an important role in the formulation of a problem. The OAF framework describes objects in terms of attributes supporting the function and the attribute affected by the function. The function is represented in a specific format and it is referred to as "OAF statement". For a paper weight, an OAF statement can be "the *mass* of **earth** and the *mass* of **paper weight** combine to create weight of **paper weight**", where the texts in italic, in bold and underlined denote the attribute, the object and the function, respectively (Sickafus, 1997).

Unlike the TRIZ, publications about the USIT in English have been sparse. Sickafus (1997) developed the USIT at the Ford Motor Company in the 1990s, which shows that the method was used in industry. "Closed-World" method and "Particle" method of the approach enable a designer to address different types of problems.

The TRIZ uses the functional analysis diagram and the substance-field functional model in the context of its relevant methods to resolve a problem, e.g. the functional analysis diagram is used to address imperfect functional performances of a system (Yang and El-Haik (2009). Tomiyama et al. (2009) point out that industry has taken a strong interest in the TRIZ, e.g. Jupp et al. (2013) and Domb and Kowalick (1998). Several researchers analysed the role of the TRIZ in different design methodologies, e.g. the design for six sigma (DFSS) (Kim et al., 2012) and product service system (PSS) design (Rovida et al., 2009). The TRIZ is commonly referred to as a method for inventive problem solving (Gadd, 2011) and its function modelling aspect has been less emphasized in the literature. There are publications on the use of the TRIZ in respect of other function modelling methods, e.g. the integration of TRIZ into Functional Basis (Nix et al., 2011) and the application of the Contact and Channel Model (C&CM) to the innovative principles of TRIZ (Albers et at, 2011a).

2.3.4.3 Statecharts

Finite-state machine is a well-known method in system modelling. State machines characterize the behaviour of a system in terms of a state at a particular time (Wright, 2005). A state machine is visually represented by a state-transition diagram (hereafter STD) (Harel, 1988). A major criticism of STDs is the representation of a system at one level, that is, they do not support top-down or bottom-up system development (Harel, 1988; Buede, 2009). The extension of STDs to statecharts by Harel (1987) with the notions of clustering, concurrency, refinement and zooming enabled statecharts to address the weaknesses of the STDs. Figure 2.16 shows a statechart of a watch focused on its alarm function.

As shown in Figure 2.16, the chart uses a rounded rectangle box to represent a state. Arrows are used to show state flows labelled with an event (or its abbreviation) and with a parenthesized condition (optional). Figure 2.16 shows the transitions between the normal displays mode and the various beeping states of the watch. The respective internal time settings of the alarms are denoted by T1 and T2, while T shows the current time. The condition P1 (and similarly for the condition P2) abbreviates “alarm 1 enabled ^ (alarm 2 disabled

∨ T1 ‡ T2).” The condition P stands for “alarm 1 enabled ∧ alarm 2 enabled ∧ T1 = T2” Harel (1987). The state “alarms-beep” and its sub-states (e.g. alarm 1 beeps) reflect the hierarchical representation of the states.

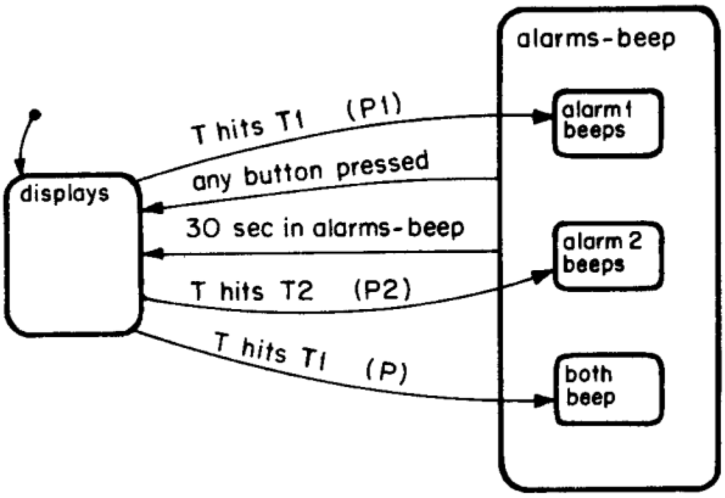


Figure 2.16: A statechart of a watch for its alarm function (Harel 1987, p. 237)

Harel (1987) noted that statecharts provide behavioural description of a system and they can be used as a stand-alone approach in this manner. They can also be used in the modelling of other aspects of the systems, for instance, data-flow specification. Statecharts provide the basis for graphical modelling languages Unified Modelling Language (hereafter UML) / SysML state machines (Weilkiens, 2006) and Simulink stateflow charts (Alur et al., 2008). Buede (2009) points out that semantics and syntax of statecharts are limited in the modelling of complex systems.

2.3.4.4 Contact and Channel Approach

Contact and Channel Model (C&CM) has been developed at the University of Karlsruhe since 1999 (Albers et al., 2004). The model went through several iterations and therefore its name has been changed as “Contact and Channel Approach (hereafter C&C^2-A)” (Matthiesen and Ruckpaul, 2012).

Albers et al. (2010) point out that interactions between elements of a technical system lead to unexpected functions. Traditional function modelling systems (e.g. Pahl et al., 2007) consider these interactions after the generation of solutions for the fulfilment of system sub-functions. Considering this issue, the C&C^2-A maps physical structure of a system to its functionality concurrently on every level of detail (Albers et al., 2014). Keller et al. (2007, p.3) uses the term “systematic function-component mapping” to describe the relationship between a physical structure and a function during the process of modelling.

Albers and Zingel (2011) suggested that the C&C²-A can model any technical system on any level of detail based on its three basic hypotheses (see Matthiesen and Ruckpaul (2012) for the hypotheses). Albers et al. (2010) noted that the representation of a function in the C&C²-A is coherent with the input/output taxonomy of technical systems (e.g. Pahl et. al., 2007) as well as state-based representation (see Albers et al., 2011b). A function is articulated in verb-object format based on the work of Hirtz et al. (2002). As noted by the second hypothesis of the approach, a function is represented by “at least two Working Surface Pair (WSP), the connecting Channel and Support Structures (CSS) and at least two Connectors” (Matthiesen and Ruckpaul, 2012, p.1021).

A Working Surface (WS) represents part of a physical object which can interact with another physical object or its environment through input or output flows (Albers et al., 2010). A WSP consists of two WSs and it shows an interface between two physical objects, or between a physical object and its environment. The interface can be in the form of energy, material and information. A CSS provides a connection between two WPSs by transferring or/and storing exchanges between two WPSs (Albers et al., 2009a). For Matthiesen and Ruckpaul (2012, p.1021), a Connector is “relevant, reduced representation of the environment for the description of the observed function.” Further elements of the approach (i.e. Limiting Surfaces and Remaining Structures) are described in Albers et al. (2005).

By using the C&C²-A, it is possible to focus on an individual problem rather than the entire system by defining the relevant part of the system and its borders. The system boundary is drawn with related to the main function accomplishment which needs to be improved or corrected. Then, locations of special interest for the function accomplishment are determined by either starting with the design (i.e. determining all WSPs and explaining what functions they fulfil) or functions of interest (i.e. locating the WPSs and the CSSs on the known functions). In the case of functionality of the related parts needs to be clarified, the system is analysed in detail using “adaptive zoom”. The C&C²-A considers different operation modes of a system by using “sequence model” (Albers et al., 2008a and 2009a). Figure 2.17 shows the representations of the main function and a sub-function of a ballpoint pen by the C&C²-A. The functions are based on the operation mode "writing with a ballpoint pen".

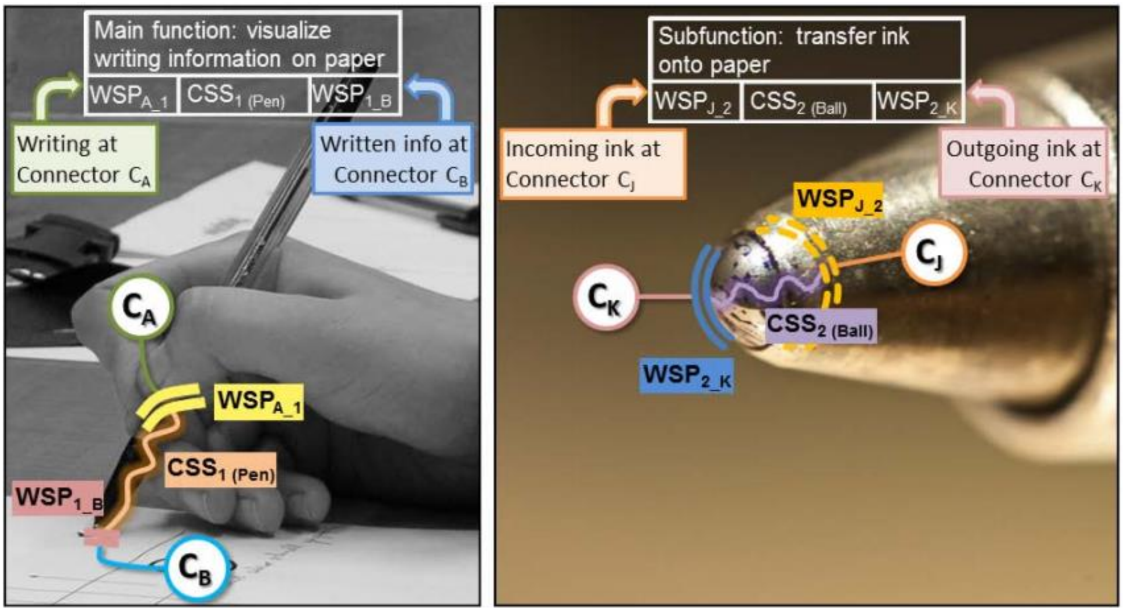


Figure 2.17: C&C²-A descriptions of a ballpoint pen (Albers and Zingel, 2011, p.4)

The C&C²-A focuses on the analysis of mechatronic (Albers et al., 2011b) and mechanical (Albers and Zingel, 2013) systems design. In terms of engineering design, the C&C²-A supports product architecture (Albers et al., 2009b), embodiment design (Albert et al., 2009a), conceptual design (Albers et al., 2010) and problem solving (Albers et al., 2008b). However, Eckert et al. (2010) point out that the practices on the C&C²-A are limited to the analysis of existing systems. There are several works in the literature on the use of the SysML for modelling the C&C²-A (Albers and Zingel, 2011; Zingel et al., 2012; Albers and Zingel, 2013).

2.3.5 Function-Behaviour-Structure-oriented approaches

2.3.5.1 Function-Behaviour-Structure framework

Gero (1990, p.28) describes the purpose of system development as "to transform function, F […], into a design description, D, in such a way that the artefact being described is capable of producing those functions". The approaches in Section 2.3.4 follow the same concept by associating the fulfilment of a function with a physical element. However, Gero (1990) discusses that there is no direct link between a function and a design description (i.e. structure, solution (S)). The Function-Behaviour-Structure (hereafter FBSt) framework proposed by Gero (1990) introduces the concept of behaviour to describe a structure (S) that implements the required function (F). Gero (1990, p.28) quotes function description of Bobrow (1984): “the relation between the goal of a human user and the behaviour of a system". Srinivasan et al. (2012) note that a combination of verbs, nouns and adjectives is used in function articulation. Behaviours are distinguished between expected behaviour (i.e. derived from the required function) and structure behaviour (i.e. derived from the structure). Structure is described as “the artefact`s elements and their relationships” Gero (1990, p.28). Figure 2.18 shows the FBSt model.

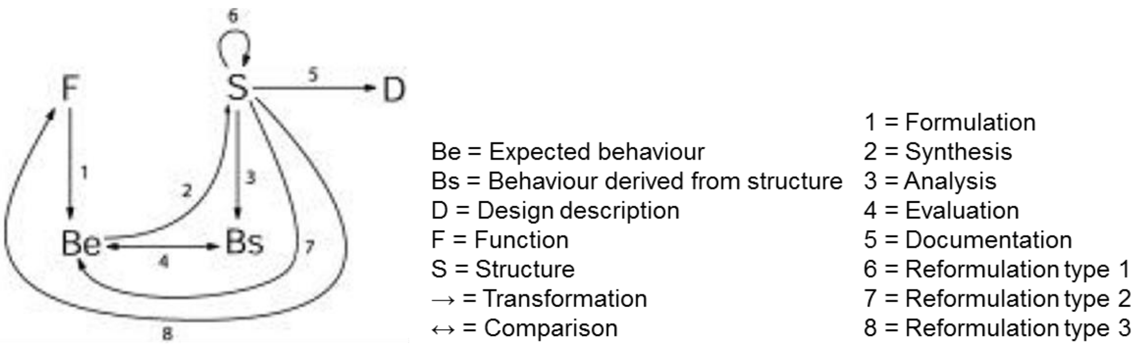


Figure 2.18: The FBSt framework (adapted from Gero and Kannengiesser, 2002, p.90)

The framework claims that there are eight fundamental processes for any design activity, as shown in Figure 2.18. The first stage (process 1) transforms the required function (F) into an expected behaviour (Be) that is expected to enable the fulfilment of the F. This follows (process 2) the transformation of the Be into a solution element (S) to exhibit the Be. The next step (process 3) is to derive the actual behaviour (Bs) from the S and then the evaluation process (4) compares the Bs with the Be to evaluate whether the S fulfils the F. If it fulfils, the design description (D) can be produced (process 5). If the S does not address the F, reformulation of the design state space (i.e. S) is addressed in terms of changes in structure variables (process 6), behaviour variables (process 7) and function variables (process 8) (Gero and Kannengiesser, 2002).

Kruchten (2005) discusses the use of the FBSt in software design. Dorst and Vermaas (2005) provide a critical analysis of the FBSt with a focus on the terms function, behaviour and structure.

2.3.5.2 Function-Behaviour-State model

Umeda et al. (1990) argue that not all functions can be described in terms of input and output of material, energy and information. Unlike Gero (1990), they introduce the notion of behaviour to support the representation of a function. Umeda at al. (1996, p.276) consider that a function cannot be represented independent of behaviour so they represent a function as an association of “to do something” and “a set of behaviours” that exhibit this function. In the Function-Behaviour-State (hereafter FBS) model, Umeda et al. (1990, p.183) describe a function as “a description of behaviour abstracted by human through recognition of the behaviour in order to utilize it”. A function is expressed in verb-object-modifier format. Here, the object denotes an entity related to the function and the modifier qualifies the function (e.g. fast). Behaviour is defined as sequential state transitions over time where a state consists of three elements; entities, attributes of entities and relations between entities. The terms state and structure are called altogether state in the FBS model (Umeda at al., 1996). Figure 2.19 shows relationship among function, behaviour and state.

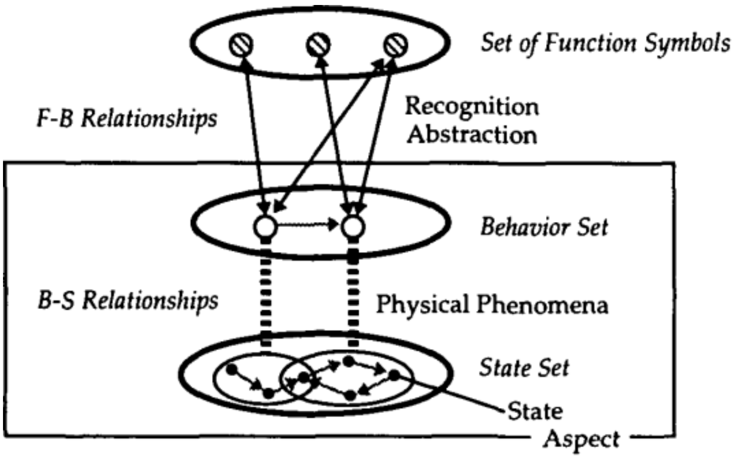


Figure 2.19: Relationship among function, behaviour and state (Umeda at al., 1996, p.277)

Physical phenomena regulates the changes of entity attributes by relating them to physical laws (Alvarez Cabrera et al., 2009). Umeda et al. (1990) suggest that the relationships between functions (F) and behaviours (B) are subjective, namely they can be described in different ways. They call these relationships F-B relationships. As shown in Figure 2.19, the representation of a function includes F-B relationships and a set of function symbols standing for human intention. The relationships between behaviours and states (S) (i.e. B-S relationships) are objective, since the behaviours of an entity are related with a set of physical phenomena from its initial state. Umeda et al. (1996) introduce the notion of aspects to represent different behaviours of the same entity based on its physical situation. An aspect is a collection of relevant states and physical phenomena of the current physical situation, as illustrated in Figure 2.19. Figure 2.20 shows an illustrative FBS model for a single function “cool down”.

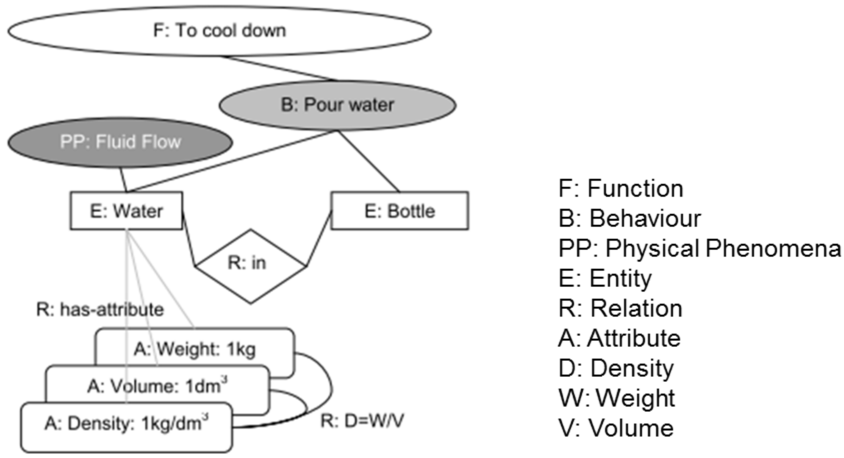


Figure 2.20: The FBS model for the function “cool down” (adapted from Van Beek and Tomiyama, 2009, p.3)

In Figure 2.20, the entities “Water” and “Bottle” have the relation “in” which means that the water is in the bottle. This relation is related to the physical phenomena “fluid flow”. “Water” has some attributes that have values, e.g. “Weight: 1kg”. The attributes are also related with each other. The function “cool down” may be decomposed into sub-functions by use of causal decomposition or task decomposition Umeda et al. (1996).

Van Beek and Tomiyama (2009) note that the FBS is particularly suitable for the analysis of existing systems. Van Beek and Tomiyama (2009) and Alvarez Cabrera et al. (2009) underline the use of the FBS in mechatronic products. Observations on the use of the FBS in industry point out that the model is very complex, since it consists of too many nodes and edges (Van Beek and Tomiyama (2009). The development of the FBS in SysML aims to address to reduce modelling efforts (Alvarez Cabrera et al., 2009).

2.3.5.3 Structure-Behaviour-Function model

Goel and Stroulia (1996) argue that the development of new designs by adapting existing designs is a common method for the conceptual design step of the product design. In contrast to the FBS and the FBSt, the Structure-Behaviour-Function (hereafter SBF) model of Goel et al. (2009) uses the concept of behaviour to map structural elements of a known device to functions. The model has been continuously adapted and developed since (see Goel, 2013).

In the SBF model, Goel et al. (2009) represents structure in terms of components, the substances held in the components and relationships among the components. The values of the parameters of a substance and/or a component can change through time. The term “state” is used to specify this change. A behaviour represents the change in substances and/or components in terms of a sequence of state transitions. Each state transition is annotated by the reasons for the transition, e.g. physical laws. A function is represented as a schema that contains a reference to the behaviour that achieves the function and specifies under which conditions (e.g. pre/post-conditions) the behaviour accomplishes the function (Goel et al., 2009). Figure 2.21 shows function and behaviour of a gyroscope which is used in gyrocompasses on ships.

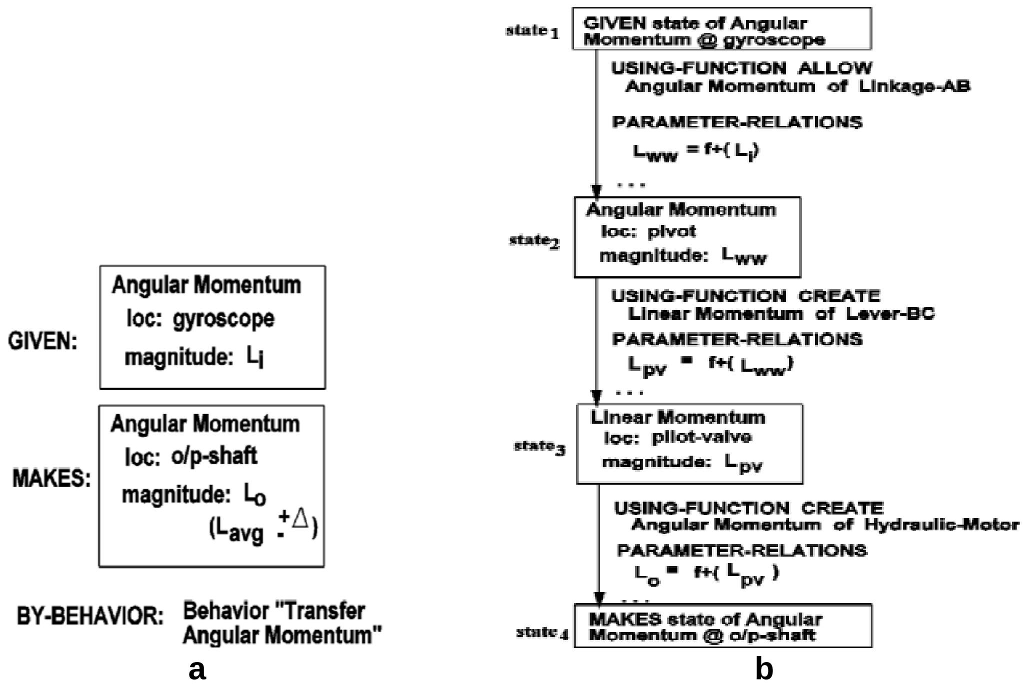


Figure 2.21: Function (a) and Behaviour (b) of a gyroscope (Goel et al., 2009, p.24-25)

Figure 2.21-a shows that the gyroscope transforms the given state (i.e. an angular momentum of specified magnitude in clockwise direction at the input location) to the output state (a proportional angular momentum of specified magnitude in clockwise direction at the output shaft location) with reference to the behaviour “Transfer Angular Momentum”. Figure 2.21-b details this transformation in terms of a set of state transitions. While the SBF expresses a function in verb-noun format (DANE, 2015), the functional context in Figure 2.21-b is shown by the annotation “USING-FUNCTION”. For example, transition from state 3 to state 4 shows that the transition take places through the function “create angular momentum” of hydraulic-motor.

2.3.5.4 Object-Process Methodology

Dori (2002) places the emphasis on the terms “behaviour” and “structure” by suggesting that they are two major aspects of any system and cannot be considered separately in the systems modelling. A combination of behaviour and structure enables a system to function (Dori, 2002). The Object-Process Methodology (hereafter OPM) of Dori (2002) describes function, structure and behaviour of a system in a single model through its basic elements objects and processes. An object represents the concept of structure in the OPM, while a process is related to the concept of function, as the object can only be changed by the process (Osorio et al., 2011). Ahmed and Dori (2009) note that the behaviour is manifested in connection with interactions between the object and the process. These interactions can take place in three ways: a process can transform an object; an object can enable a process, and an object can trigger an event that invokes a process. For Dori (2002), function of a system influences its structure and its behaviour and it is described as “an attribute of object that describes the rationale behind its existence, the intend for which it was built, the purpose for which it exists, the goal it serves, or the set of phenomena or behaviours it exhibits” (p.251).

The key difference of the OPM to the reviewed approaches in the previous sections is dual-expression of a system graphically and textually through Object-Process Diagram (hereafter OPD) and Object-Process Language (hereafter OPL). The OPL is the textual counterpart of the OPD - in other words, each OPD element is articulated as an OPL sentence (Dori and Reinhartz-Berger, 2003). In the OPL, a function sentence starts by listing processes followed by objects, to which the phrase “function to” is added, followed by the function name in bold italic Arial font. For example, “**Moving** and **Car** function to ***enable translation***” (Dori, 2002, p.270). Figure 2.22 shows OPD template and OPL script for a generic function and a generic system architecture (see Dori (2002) for OPD diagram symbols and OPL sentence structures).

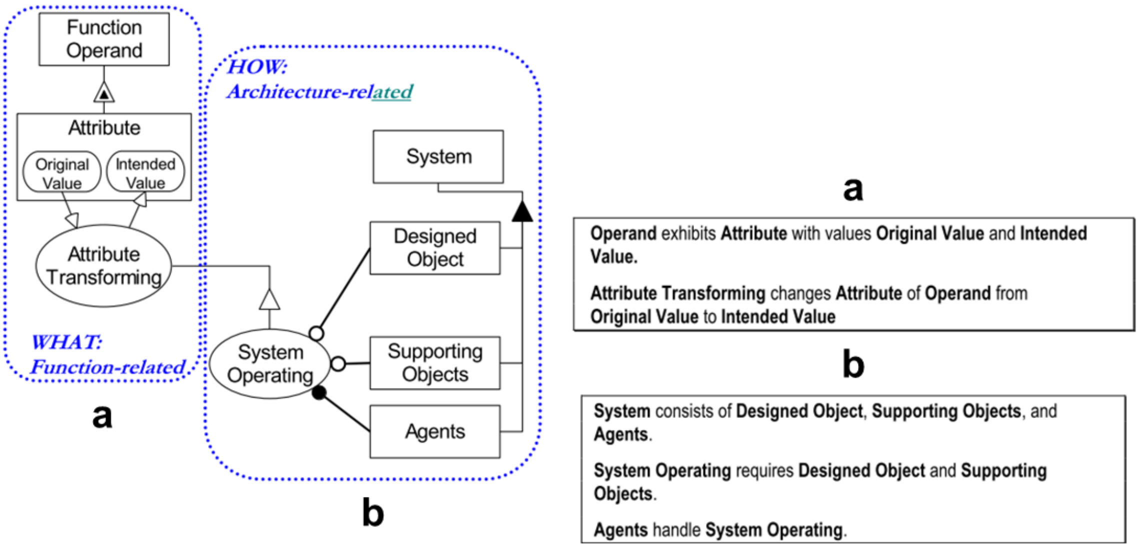


Figure 2.22: OPD template and OPL script (adapted from Soderborg et al., 2002, p.3-4)

As shown in Figure 2.22, Dori (2002) distinguishes between function and architecture (behaviour/structure combination) by aligning them with the questions “What result do you desire?” and “How does the system achieve it?”, respectively (Soderborg, 2002). According to Osorio et al. (2011), the modelling process in the OPM starts by identifying the intended function in relation to the utility of the system to the users. This follows the description of the operands and value attributes in respect of the function. Once the operators of the system are identified, the model may be decomposed for further analysis. Ahmed and Dori (2009) point out that the OPM enables analysing a system to any level of detail by offering three complexity management mechanisms: unfolding/folding, in-zooming/out-zooming, and state expressing/suppressing. By a combination of these mechanisms, the decomposition of a system’s function and structure can be represented in a top-down manner (Osorio et al., 2011).

The OPM has been used mainly in product design and systems engineering (Osorio et al., 2011). Peleg and Dori (1999) note that the methodology is applicable to both system analysis and system design. Howes (2008) suggests that the OPM supports the inclusion of human activities in systems modelling by representing them as agents or objects. Grobshtein and Dori (2011) introduce a comparison between the OPM and the SysML.

2.3.6 Other approaches

Section 2.3.2-to-2.3.5 provided a review of function modelling approaches with a strong focus on their generic characteristics, i.e. function definition, function articulation, function representation and function decomposition. A number of approaches to support particular characteristics of function modelling may be found in the literature.

2.3.6.1 Function Definition

Several researchers proposed different types of function definitions.

- Simon (1996) thinks of an artefact as an interface between an inner environment (i.e. organization and substance of artefact) and outer environment in which the artifact operates. The artefact fulfils its intended function if its inner environment matches its outer environment, or vice versa.
- Chandrasekaran and Josephson (2000) distinguish between device- and environment-view of functions. The former defines a function in terms of components of a device, while the latter focuses on the effect of the device on the environment in which it is located and this view of function is also referred to as “function as effect”. Chandrasekaran and Josephson (2000) introduce the terms “mode of deployment” and “role” for the description of the use of the device to produce the intended effect and for the description of the effect of the device on its environment, respectively.
- Deng (2002) distinguishes functions in the upper and lower levels of a system by introducing purpose and action functions. The former describes the intention of the designer and it is human oriented, while the latter refers to an abstraction of intended behaviour to be exhibited by an artefact and it is human related. The action functions support the fulfilments of the purpose functions.

- Kitamura and Mizoguchi (2010) propose more detailed classification of functions by making a distinction between actual-capacity function, artefact-device function and essential-accidental function.
- Crilly (2012) extended the concept of Chandrasekaran and Josephson (2000) by introducing endogenous and exogenous functions.

2.3.6.2 Function Articulation

Keuneke (1991) describes a function as the intended purpose of a device. Like the FBS and the SBF, the device achieves its function through the causal sequence of partial state/predicate transitions. However, Keuneke (1991) suggests articulating a function in “to do” form addressing relevant operations on states, i.e. ToMake, ToMaintain, ToPrevent and ToControl. Deng (2002) suggests that syntactic representation of a function can be in the form of a sentence or a mathematical formulation.

2.3.6.3 Function Representation

Petri nets (hereafter PN) follows a similar methodology to task-oriented approaches in Section 2.3.2 by modelling a system in terms of discrete events. However, the PN has a very different system representation style. Places, transitions, arcs and tokens are the main elements of the method. A circle denotes a place and it represents an input or an output of an activity in a process. A transition represents an action in the process and it is denoted by a bar. An arc is an arrow that indicates an input to a transition by pointing from a place to transition, and vice versa. A token is denoted by a dot within the place, which shows the presence of the object represented by the place. The PN use the term “fire” to refer to a function. If each place in a process (that has an arc pointing to the transition) possesses a token, a transition can fire. Once the transitions take place, the tokens on the inputs are removed, and new tokens are placed on the outputs (O`Donovan et al., 2005). Murata (1989) provided an overview of the PN, who also pointed out that the PN can equivalently represent finite state machines. There are many extensions of the PN (Buede, 2009). Mackenthun et al. (2001) represent the use of the PN based on state-, event- and object-oriented approaches. While O`Donovan et al. (2005) note that the PN have been used for modelling a wide range of systems that cross engineering domains, Buede (2009) argues that the PN are very sophisticated to be used in the field of engineering design.

A bond graph only models the energy flow through a system by using a set of elements (i.e. ports) which are associated with the variables “effort” and “flow” (McBride, 2005). While bond graphs support the analysis of complex systems across disciplines (Summers et al., 2001), Triengo and Bos (1985) note that it is difficult to model mechanical systems. Umeda et al. (1990) point out that the model focuses on structure and behaviour of a system, but not on its functions. Specific symbols are used in the modelling of a system (see McBride, 2005).

2.3.6.4 Function Decomposition

Eckert (2013) notes that there are four approaches to function decomposition in industry: top-down, important things first, issue driven and power flow throughout the system. The approaches reviewed in this chapter so far show that top-down decomposition is common in literature. In addition to the reviewed approaches, there are different approaches to top-down function decomposition.

- “Divide-and-conquer” (hereafter D&C) approach (Chmarra et al., 2008) decomposes a problem into sub-problems until they can individually be solved. The individual solutions for these problems are combined into a single solution for the main problem. The D&C approach is used by various function modelling approaches, e.g. Pahl and Beitz method in Section 2.3.3.1. Komoto and Tomiyama (2011) extended the D&C approach to establish a theory of decomposition in conceptual design. They argue that the D&C approach is not well-formalized, e.g. there is no unique method to decompose a system.

- Koopman (1995) presented a taxonomy of decomposition strategies based on structures, behaviours, goals and their combinations. For example, Koopman (1995) suggests that the axiomatic design (Suh, 1990) is a variation of combined structure-goal decomposition, since it carries out goal and structure decomposition concurrently.

- Crilly (2010) introduces the notion of “nested systems” by nesting the systems within each other in a systems hierarchy.

2.3.6.5 Function Modelling using System Modelling Language

As mentioned in Section 2.3.2.1, SysML is a graphical modelling language. Many function modelling approaches reviewed in the previous sections aim to promote their practical applicability by exploiting SysML. According to Zingel et al. (2012), SysML is the most popular modelling language in systems engineering design. SysML includes many diagrams, as shown in Figure 2.23.

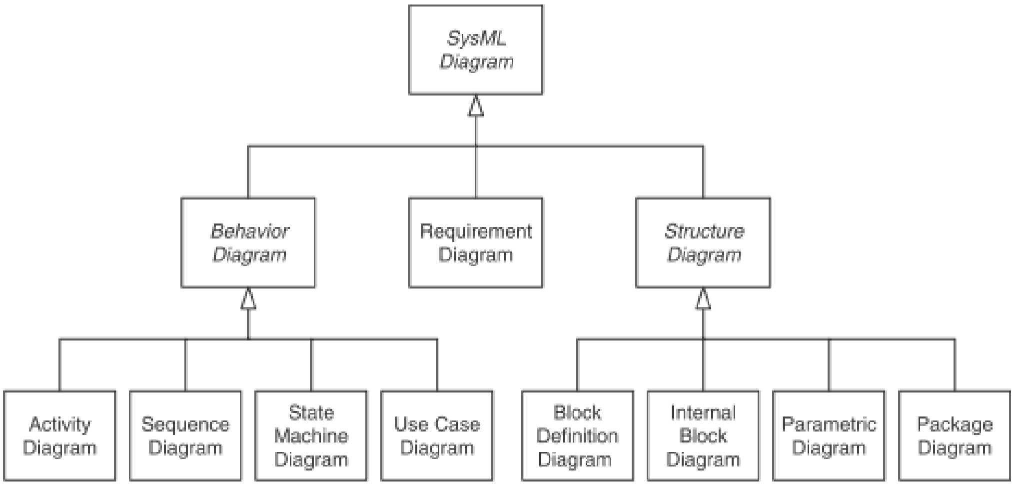


Figure 2.23: SysML diagrams (Friedenthal et al., 2008, p.30)

Of SysML diagrams in Figure 2.23, behaviour diagrams are relevant to function modelling since they represent functionality in terms of how a system operates.

2.3.6.5.1 Use Case Diagram

Use cases describe the functionality of a system. A use case is achieved through actors which may be human or other external entity. Actors interact directly with the use cases and indirectly with each other. Use cases are represented in a use case diagram (Friedenthal et al., 2008). Figure 2.24 represents an excerpt from a use case diagram for the use case “operate vehicle” of a vehicle.

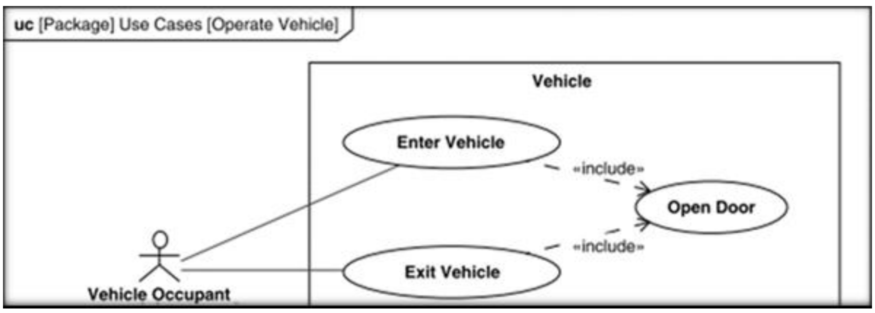


Figure 2.24: An excerpt from a use case diagram for the use case “operate vehicle” (Friedenthal et al., 2012, p.59)

Figure 2.24 shows that the use cases “enter vehicle” and “exit vehicle” within ovals are related to the actor “vehicle occupant” which is shown as a stick figure with the actor`s name below (as shown in Figure 2.24) or as a rectangle containing the word “actor” with the actor`s name underneath. Actors and use cases are connected by association paths, denoted by lines. A dashed line with an open arrow at the sub-use case “open door” shows the link between the sub-use case and its use case. The use cases “enter vehicle” and “exit vehicle” require the use case “open door” (Friedenthal et al., 2008).

Relationships between a system and its actors in respect of a particular use case can be analysed in detail using sequence diagram, activity diagram and state machine diagram.

2.3.6.5.2 Sequence Diagram

Message-based interactions between the system and the actors can be mapped by sequence diagrams. Figure 2.25 shows a sequence diagram for “Turn On Vehicle” interaction for a vehicle.

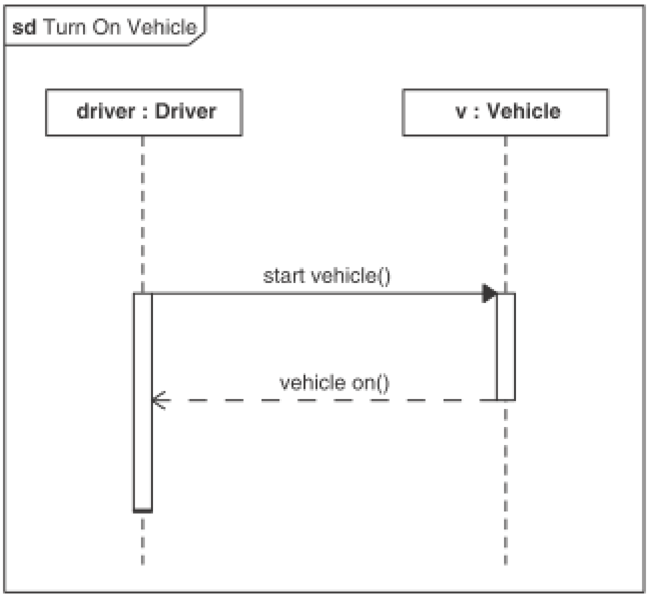


Figure 2.25: Sequence diagram for the “Turn On Vehicle” interaction (Friedenthal et al., 2012, p.62)

Figure 2.25 represents the actor “driver” and the system “vehicle” in rectangles. A lifeline represents the relevant lifetime of the actor and the system and it is shown in Figure 2.25 with dashed lines descending from the base of the rectangles with respect to time. A synchronous message in sequence diagram denoted by a closed arrowhead and it is commonly accompanied by a reply message which is denoted by an open arrowhead with dashed lines. As shown in Figure 2.25, the synchronous message from the Driver shows that the Driver request the Vehicle to start, and the Vehicle responds this request with the reply message "vehicle on". While sequence diagram focuses on the exchanges of messages, the passage of material and energy can also be indicated on the diagram in parentheses after the message name (Friedenthal et al., 2008).

2.3.6.5.3 Activity Diagram

Activity diagrams are useful where interactions between the system and the actors include the flow of inputs, outputs and control. An excerpt from an activity diagram in Figure 2.26 shows the flow of activities between the actor "driver" and the system "vehicle" regarding the control of power of a vehicle.

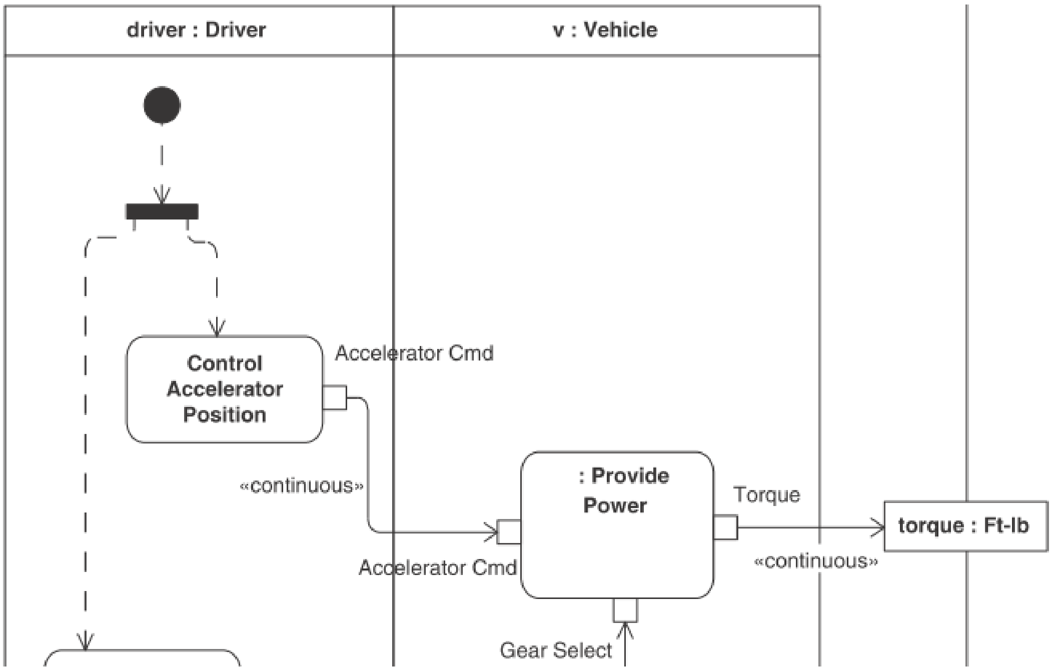


Figure 2.26: An excerpt from an activity diagram regarding the control of power of a vehicle (Friedenthal et al., 2012, p.63)

The activities "control accelerator position" and "provide power" are shown in rectangles with rounded corners in Figure 2.26. A SysML activity adapts token-based semantics of Petri-Nets. The input and the output values of inputs, outputs and control correspond to tokens on the activity and they may represent information, matter or energy. Tokens are placed on pins which are represented as small rectangles. Figure 2.26 shows that the activity "provide power" processes the input tokens “accelerator command (cmd)” and “gear select” on its input pins and generates the output token “torque” on its output pin (Friedenthal et al., 2008).

2.3.6.5.4 State Machine Diagram

State machine diagrams may be required where interactions between the actors and the system cannot easily be represented in terms of an ordered sequence of events. Figure 2.27 shows an excerpt from a state machine diagram for drive vehicle states.

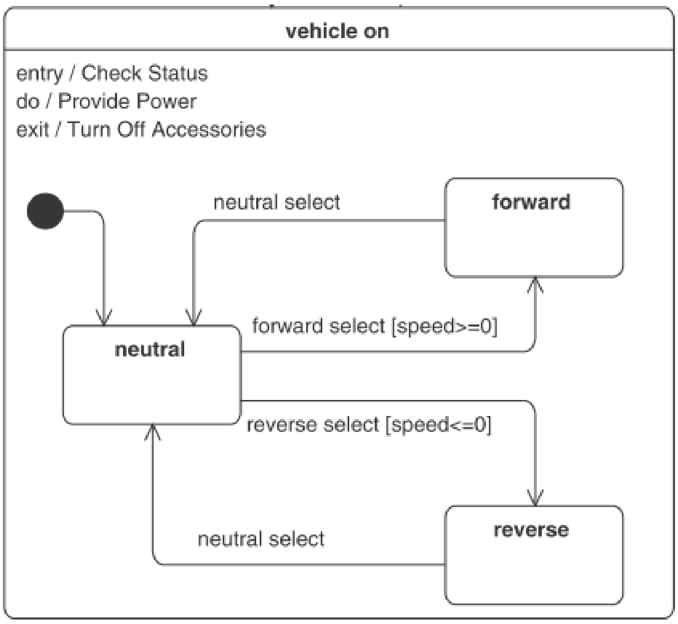


Figure 2.27: An excerpt from a state machine diagram for drive vehicle states (Friedenthal et al., 2012, p.65)

Figure 2.27 shows that the state of vehicle is “vehicle on” which has the sub-states “forward”, “neutral” and “reverse”. The diagram shows a transition between the sub-states by an arrow with the trigger`s name and the guard condition (in square bracket). The terms “entry”, “do” and “exit” denote “entry behaviour”, “do behaviour” and “exit behaviour” of the state “vehicle on”, respectively.

The reviewed SysML diagrams in this section focused on notations relevant to the presented examples (see Friedenthal et al. (2012) for the rest of the notations). The choice of which diagram to use is at the designer`s discretion (Friedenthal et al., 2008). For example, an activity diagram for a use case may be more useful than a sequence diagram for the same use case. Behaviour diagrams of the SysML do not put great emphasis on how defining, articulating and decomposing functions. They focus on reducing modelling efforts by introducing a strong formalism (Eisenbart et al., 2015).

2.4 Critical Review of Function Modelling Approaches

Section 2.3 reviewed function modelling approaches with a strong focus on four criteria: function definition, function articulation, function representation and function decomposition. Table 2.1 provides a succinct review of the approaches based on these criteria.

Table 2.1 A review of function modelling approaches in Section 2.3

Approach (Reference)	Function Definition	Function Articulation	Function Representation	Function Decomposition
FFBD (NASA, 2007)	action	action verb + a noun phrase	block	Each block in the first level of the diagram is expanded to a series of functions and so on.
FAST (Kaufman and Woodhead, 2006)	intent or causal action	active verb + measurable noun	block	How-why-when questions are applied to each function.
DSM (Eppinger and Browning, 2012)	N/A	based on designer`s discretion	cell	N/A
IFM (Eisenbart, 2014)	intended behaviour	verb + noun format or sentence	block	The sub-processes (i.e. sub-functions) are modelled in a separate process flow view.
Pahl and Beitz (Pahl et. al, 2007)	intended input-output relationship in terms of the flows of energy, material and signal	verb + noun	block (i.e. black box)	The overall function is broken down into sub-functions.
Ullman (Ullman, 1992)	intended input-output relationship in terms of the flows of energy, material and information or states	action verb + a noun phrase	black box	The overall function is broken down into sub-functions.
Ulrich and Eppinger (Ulrich and Eppinger, 2003)	input-output flows of energy, material and signal	verb + noun	black box	The overall function is broken down into sub-functions.

FB (Stone and Wood, 2000)	input-output flows of energy, material and signal	action verb + noun	black box	The overall function is broken down into sub-functions.
IDEF (Buede, 2009)	intended input-output relationship	verb + noun phrase	box	The top-level function (i.e. overall function) is decomposed into its sub-functions
TRIZ Functional Analysis Diagram (Yang and El-Haik, 2009)	the flow of an action from the source of the action to the action receiver	verb	arrow	N/A
TRIZ Substance-Field Functional Model (Fey and Rivin, 2005)	an interaction between two elements of a system	N/A	a triangle whose corners represent substances and a field	N/A
USIT OAF framework (Sickafus, 1997)	the relationship between object attributes supporting the function and object attribute affected by the function	verb + object attribute	OAF statement	N/A
Statecharts (Harel, 1987)	transition between states	verb + noun format or sentence	arrow	zooming in/out
C&C^2-A (Matthiesen and Ruckpaul, 2012)	input-output of the flows of energy, material and information or states	verb + object	at least two Working Surface Pair, the connecting Channel and Support Structures and at least two Connectors	adaptive zoom
FBSt (Gero,1990)	the relation between the goal of a human user and the behaviour of a system	a combination of verbs, nouns and adjectives	text	N/A
FBS (Umeda et al., 1990)	transition between states	verb + object + modifier	Diagrammatic representation of a function includes F-B relationships and a set of function symbols.	causal/task decomposition

SBF (Goel et al., 2009)	transition between states	verb + noun	arrow	The high level input-output state is decomposed into intermediate states
OPM (Dori, 2002)	the relationship between object attributes	OPL sentence	OPD	unfolding/folding; in-zooming/out-zooming; state expressing/suppressing

Table 2.1 is quite revealing in several ways.

- The column "Function Definition" confirms the view of Vermaas (2011, p.98): "function lacks a single precise meaning. It is a term that has a number of co-existing meanings, which are used side-by-side in engineering."
- Table 2.1 also reveals that a function can generally be defined under two concepts: transformation/flow and goal. Figure 2.28 shows the concepts in the definition of a function.

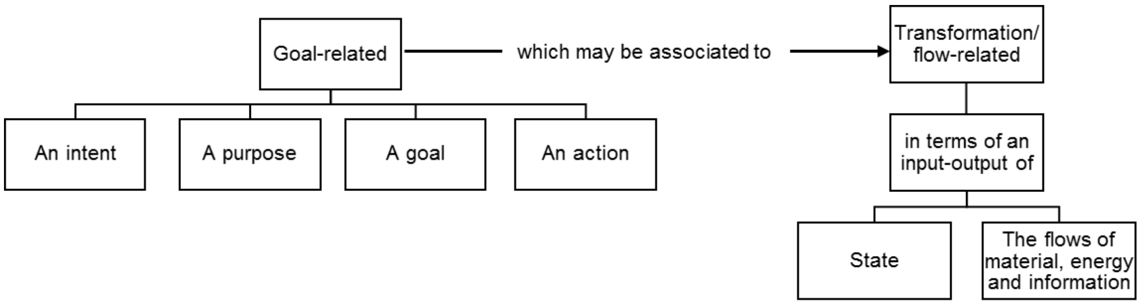


Figure 2.28: Concepts in function definition

While transformation/flow-related definition of a function is specified in terms of input-output flows of material, energy and information or state, goal-related function definition has various forms, e.g. action. Several approaches associate the goal-related concept with the transformation/flow related concept, as indicated in Figure 2.28. For example, Pahl et. al, (2007) identify the purpose of a system with respect to the intended input/output relationship of the system.

- The column "Function Articulation" in Table 2.1 shows that articulation of a function in verb-noun format is common practice in literature. The approaches define a function with respect to the transformation/flow-related concept associate the articulation of a function with the flows of material, energy and information or state.

- It is apparent from the column “Function Representation” in Table 2.1 that a function is generally represented by a black box. This enables the approaches to integrate the elements used in the definition and articulation of a function into a single model. For example, Pahl et al. (2007) define a function as “the intended input/output relationship of a system” and articulate it in “verb-noun” form. A black box model can represent these elements altogether. State-based approaches (e.g. Statecharts (Harel, 1987)) use arrows representing transition between states.
- The column “Function Decomposition” shows that there is no uniform method or algorithm for functional decomposition. It is important to note that task- (expect the DSM) and flow-oriented approaches follow the same methodology at high level (i.e. the overall function is broken down into sub-functions), however they follow different ways in the division of the overall function.

2.5 A Critique of Function Modelling Approaches

The review in Section 2.3 shows that the approaches have been introduced with the aim of developing functional model of a new product or/and an existing product. While functional model of an existing product supports the exploration for an improvement in the existing design solutions without changing the identity of the product (Kaufman and Woodhead, 2006), e.g. turbofan bread toaster, functional model for a new product may be about developing a novel technology that has not been done before (Pahl et al., 2007), e.g. automated omelette makers.

Many function modelling approaches have been illustrated on existing products based on reverse engineering (Summers et al., 2013). A notable example of this is the FB of Otto and Wood (2001). Hypothetically, these approaches should be able to represent all functional requirements during the operation of an existing product. This section sets out to determine the need for a new function modelling approach by presenting a critique of the reviewed approaches in Section 2.3 based on their ability to provide functional model of an existing system in a structured way.

In Section 2.3.1, Srinivasan et al. (2012) gave a generic picture of views of function by categorizing these views into Level of Abstraction, Requirement-Solution, System-Environment and Intended-Unintended. The views “Level of Abstraction” and “Requirement-Solution” are about addressing different level of detail of a system, while the view “System-Environment” focuses on the development of the system per se or the effect of the system on the environment in which it operates. The view “Intended-Unintended” means to represent intended and unintended functions of a system. This view can be considered as one of the basic requirement of a functional modelling approach. Because, a function modelling approach should also promote the identification and the representation of unintended by-products, which are dubbed by Johnson (2005) “emergent properties”. If a modelling approach can address the view “Intended-Unintended”, other views of function can be supported easily. Coherent with Erden et al. (2008), other basic requirement of a functional modelling approach can be described as the ability to represent the functionality of a system in terms of chains of sub-functions with relation to each other. The critique of the reviewed approaches is carried out based on these two requirements.

2.5.1 Task-oriented approaches

Task-oriented approaches in Section 2.3.2 represent functional model of a system in respect of causality, i.e. the first function is connected to the second function with respect to time, and so on. While the DSM (Section 2.3.2.3) serves to support visual aspect of function modelling, the FFBD (Section 2.3.2.1), the FAST (Section 2.3.2.2) and the IFM (Section 2.3.2.4) support top-down development of a functional model. However, these approaches do not provide a formal way of developing function chains and combining these chains with each other. They merely emphasize causal relationship between functions.

2.5.2 Flow-oriented approaches

Flow-oriented approaches in Section 2.3.3 extend the concept of task-oriented approaches by introducing the black box and the flows of material, energy and information/signal. As discussed in Section 2.3.3, they represent the overall function as a black box and decompose the black box into sub-functions in different ways. The method of Pahl and Beitz (Section 2.3.3.1) differentiates between main functions and auxiliary functions, but it is not clear how to link the auxiliary functions with the main functions. Function decomposition methodology of Ullman (2010) (Section 2.3.3.2) does not focus on the combination of chain of functions, while Ulrich and Eppinger (2003) (Section 2.3.3.3) adapt the method of Pahl et al. (2007) with a strong focus on problem decomposition. Unlike the main flow methodology of Pahl and Beitz, the FB of Stone (Section 2.3.3.4) develops a chain of sub-functions for each input flow on the black box and aggregates them into a single functional model. However, the approach does not specify a way of aggregating these chains. The development of a function model of a system based on the given inputs on its black box may prevent proper identification of unintended functions while the system is in use. For example, for a bread toaster, the moisture generated during the process of toasting bread can affect the process if the flow of moisture is not appropriately managed by the toaster device. Functional model for the toaster in Figure 2.11 does not include the flow of moisture since the moisture is not represented on the toaster black box.

2.5.3 Function-Structure-oriented approaches

As discussed in Section 2.3.4, function-structure-oriented approaches are solution oriented, that is, they provide a particular answer to how a function is achieved. A function consists of four elements in the IDEF0 (Section 2.3.4.1.1); in addition to the flows of inputs and outputs, the IDEF0 also represents controls and mechanisms on a function box. While the approach can represent the operation of a system in terms of a chain of functions in detail, it does not provide a methodology for the identification of these elements, e.g. which element should be identified first. Similarly, the TRIZ and the USIT (Section 2.3.4.2) place a great emphasis on the elements used in the description of a function (e.g. OAF framework), but the use of these elements in the development of a functional model is not detailed.

The statecharts (Section 2.3.4.3) can capture possible functions in a complex system in terms of state transitions, however it is not clear how to start developing state transitions. Top-down system development by statecharts is difficult due to the same reason. The C&C^2-A (Section 2.3.4.4) poses the same problem. Though Albers et al. (2010) noted that the approach is coherent with the input/output taxonomy of technical systems, it is difficult to put this into practice. The main reason for this difficulty is to use multiple elements in the representation of a function (i.e. at least two Working Surface Pair, the connecting Channel and Support Structures and at least two Connectors) which can easily make complicated functional model of a reasonable sized system.

2.5.4 Function-Structure-Behavior-oriented approaches

An innovation of function-behavior-state-oriented approaches is the introduction of the notion of behavior in function modelling. The FBSt framework (Section 2.3.5.1) is about deriving structures from functions through behaviors. The framework does not focus on the development of a chain of functions. The FBS model of Umeda (Section 2.3.5.2) associates functions with sequential state transitions, however the representation of a functional model can easily become complicated and bulky since each state consists of entities, attributes and relations. The SBF (Section 2.3.5.3) provides a more structured methodology as compared to the FBSt and the FBS by decomposing given input and output states into sequential state transitions. This also limits the model to the development of non-branching state transitions. One advantage of the SBF model and the statecharts compared to the FBS is to represent a state in a single representation which makes easier to illustrate a functional model. Unlike the SBF and the statecharts, the OPM (Section 2.3.5.4) has too many symbols to give a system representation. It can be difficult to integrate system functions at the same level of detail.

Key weaknesses of the reviewed approaches can be summarized as follows:

1) While the reviewed approaches provide different ways of developing function model of a system, the proposed guidelines are insufficient to support function modelling of a system in a structured way. For example, it does not sound clear how to combine different chains of functions in the flow-oriented approaches.

2) The approaches do not seem to consider possible unintended by-products generated while the system is in use, e.g. functional model of a system by the functional basis is developed based on the inputs on the black box; therefore, the model for the toaster in Figure 2.11 does not include the moisture generated during the process of toasting.

3) The review of the modelling approaches in Section 2.3 verifies the observation of Liu et al. (2015) on current state in function modelling. Most of the approaches focus on the modelling of one mode of operation of systems through analyzing the overall system function.

4) It is noteworthy that the more functional elements are included in a framework, the more function model of the system becomes complicated. For example, the C&C²-A (Matthiesen and Ruckpaul, 2012) represents a function by using at least five elements, i.e. pairs of Working Surfaces, the connecting Channel and Support Structure and Connectors.

These findings and the described problems with the existent function modelling approaches in literature suggest that there is a gap in function modelling research, which is given in Figure 2.29.

There is still a need for development of a function modelling approach, to ensure that all flows regarding intended-unintended functions through complex multidisciplinary systems with multiple operation modes are captured in a structured way.

Figure 2.29: Research gap

This thesis aims to address the research gap in Figure 2.29 by developing a function modelling approach based on the SSFD. Section 1.2 summarized the current state of the SSFD and listed some key limitations of the SSFD. As discussed in Section 1.2, the SSFD supports system design and decomposition on a functional basis within a multidisciplinary environment, however it lacks structured definition of the key elements of “state” and “function” and the current SSFD does not provide a structured guideline on the development of functional model of complex multidisciplinary systems with multiple operation modes.

The following chapters introduce the development of the SSFD by addressing its limitations in a coherent way.

2.6 Chapter summary

This chapter started by describing the need for function modelling in engineering design. An overview of function modelling approaches was provided based on four perspectives highlighting issues of practical relevance.

The summary of the key findings revealed common characteristics of the reviewed function modelling approaches. The critique of the reviewed approaches clarified the gap in the research.

3. Function Modelling based on System State Flow Diagram

The development of the SSFD framework in this thesis is represented in three chapters by mapping the development of theory (i.e. deduction) and the validation of the proposed theory (i.e. induction) iteratively, as represented in Section 1.4. This chapter is about the first leg of the framework development. The chapter starts by introducing the key concepts and elements of the SSFD framework on the basis of a critical analysis of other functional modelling frameworks. The proposed elements are tested on the representation of function chains of an in-tank fuel delivery system. Next, the key research questions that arise from this application are phrased, before concluding the chapter with a methodology followed in the next stages of the framework development.

3.1 System state flow diagram as a framework for function modelling

3.1.1 The basis for the system state flow diagram

The review in Section 2.3 showed that an engineered system is commonly represented as a black box (e.g. flow-oriented approaches in Section 2.3.3), showing the inputs and the outputs of the system in terms of the flows of material, energy and information, as illustrated in Figure 3.1.

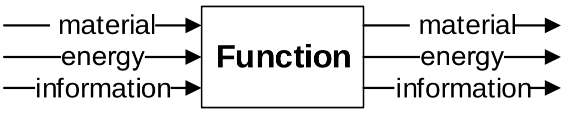


Figure 3.1: Black box

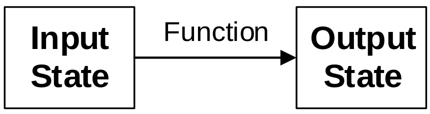


Figure 3.2: State transition

Vermaas (2009) mentions that the main weakness of this representation is that it is based on the assumption that every function has both input flows and output flows of material, energy and information. While flow-oriented approaches in Section 2.3.3 represent the flows of material, energy and information through the system in terms of inputs and outputs, representation of some functions using the methodologies of these approaches may create misleading information. For example, Stone and Wood (2000) describe the function “stop” as “to cease the transfer of the flow of material or energy” and therefore representing an output flow for this function may be contradictory and confusing. This is due to the fact that flow-oriented approaches do not specify the location of the input flow and the output flow as well as their characteristics in the representation of the flow. While Tate (1999) tried to address this problem by representing an operand with attributes at a particular time, the work of Tate (1999) did not focus on the development of function chains.

Section 2.3 also pointed out that a function can also be represented in terms of a state transition. Chapter 2 reviewed a variety of state-based function modelling approaches. As mentioned in Section 2.3.4.3, the statecharts are the driving force behind the UML and the SysML state machine diagrams (Weilkiens, 2006), which are popular tools for model-based development of multidisciplinary systems (Albers and Zingel, 2013). Coherent with the general principles of the statecharts, flow-based representation in Figure 3.1 can be represented graphically as a state transition in Figure 3.2 in which, by convention, a state is denoted by a box and a function is represented by an arrow which is required to transfer between states (Campean et al., 2013b).

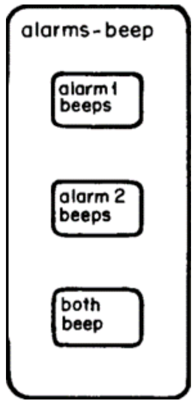
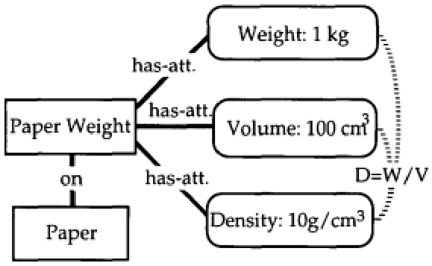
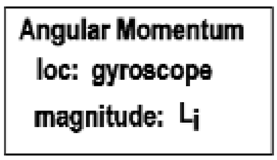
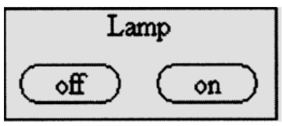
The use of states in function modelling raises questions regarding the definition of a state and the articulation of a function in related to a state transition. These questions will be addressed in the next sections.

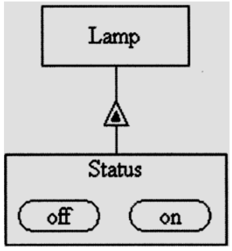
3.1.2 Analysis of the basic constituents of a state-based diagram

3.1.2.1 Graphical conventions for state definitions and representations

The review of function modelling approaches in Chapter 2 showed that the statecharts (Harel, 1987), the FBS (Umeda et al., 1996), the SBF (Goel et al., 2009) and the OPM (Dori, 2002) use states in function modelling of a system, while the method of Ullman (2010) and the C&C²-A (Matthiesen and Ruckpaul, 2012) are compatible with state-based representation. Table 3.1 shows how the reviewed function modelling approaches define and represent a state with an example for a better understanding.

Table 3.1: Definition and representation of a state in the reviewed function modelling approaches

Approach (Reference)	State Definition	State Representation	Example
Statecharts (Harel, 1987)	the condition of a given element which can be specified in terms of a set of value combinations	A state is represented by a rounded rectangle at any level. A sub-state is expressed by using the notion of encapsulation, e.g. the state "alarm 1 beeps".	Alarm function of a Watch (p.237) 
FBS model (Umeda et al., 1996)	The state of an entity is described in terms of a set of attributes that have values and relations among relevant entities	Illustrative example on the right represents an entity, an attribute and a relation as a rectangle, rounded rectangle and line, respectively.	Paper weight (p.276) 
SBF model (Goel et al., 2009)	the values of the parameters of substances and/or components	The state of the substance "angular momentum" is shown in a rectangle box along with its parameters (e.g. location) and the values of these parameters (e.g. gyroscope)	Gyroscope (p.24) 
OPM (Dori, 2002)	the situation of an object at a particular time	**Implicit representation:** The object "lamp" is shown inside a rectangle box. , while the state of the lamp is represented by a rounded rectangle box within the object, e.g. off	Lamp (p. 85) 

		Explicit representation: The object “lamp” in the rectangle box exhibit the attribute “status”. The value “on” and “off” of “status” within rounded rectangle boxes are the lamp`s states.	

As shown in Table 3.1, the reviewed approaches generally represent a state as a box which may be rectangle or rounded rectangle. The FBS of Umeda et al., 1996) is an exception since it uses multiple elements in the representation of a state. The representation of Harel (1987) is prevalent in the development of multidisciplinary systems, e.g. the SysML state machine diagram (Weilkiens, 2006). Therefore, the SSFD adopts the state representation of Harel (1987) by representing a state as a round-cornered box containing the name of the state in bold, as shown in Figure 3.3.

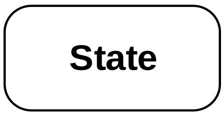


Figure 3.3: Statechart state representation adopted by the SSFD

In the case of state definition, it is apparent from Table 3.1 that the reviewed approaches associate the term “state” with the situation of a given element (Harel, 1987), an entity (Umeda et al., 1996), substances and/or components (Goel et al., 2009) and an object (Dori, 2002). Umeda et al. (1996) and Dori (2002) point out that the situation of an entity and an object can be specified in terms of a set of attributes that have values, while Goel et al. (2009) use the term “parameter” instead. Further investigation into these terms needs to be done to establish the definition of a state for the SSFD. The following sub-sections discuss the issue of state definition on the basis of these terms.

3.1.2.1.1 State Definition via Objects

The approaches in Table 3.1 use the terms “element”, “entity”, “substance”, component” and “object” as the basis for the definition of a state. The terms “Entity”, “Component” and “Object” are used interchangeably in literature. Umeda et al. (1996) describe an entity as a component. Unlike Umeda et al.

(1996), Alvarez Cabrera et al. (2009) suggest that an entity corresponds to an object. Goel et al. (2009) use the term “Substance” to describe structure of a system. The substances are contained in the components. Weilkiens (2006) suggests using the term “element” in the UML/SysML state machine diagrams.

The term “object” is used in the definition and the articulation of a function by numerous function modelling approaches reviewed in Chapter 2 (see Table 2.1 in Section 2.4). There is a variety of definitions of the term “object” in literature. Stone and Wood (2000) describe it as “the recipient of a function`s operation.” Sickafus (1997) defines an object as a “tangible item”. For Fey and Rivin (2005), it is a component of the system that is to be controlled, processed or modified.

Object is one of the main elements of the OPM and therefore Dori (2002) puts great emphasis on this term. According to Dori (2002, p.57), “an object is a thing that has the potential of stable, unconditional physical or mental existence.” This description places importance on two aspects of an object which are dubbed by Dori (2002) “physical existence” and “mental existence.” The former pertains to tangible aspect of an object which can be seen, touched and experienced. Dori (2002) calls this type of object “physical object” which consists of matter and obeys the basic laws of physics, e.g. a stone. Mental existence of an object is referred to as being intangible. It is called by Dori (2002) “informatical object”. The laws of physics do not apply to this type of object which can be apprehended depending on its form such as being recorded on some tangible medium that can be some electromagnetic medium, paper, the human brain, etc., for example, a childhood memory (Dori, 2002).

Stone and Wood (2000) describe a flow as the object of a sub-function. Their flow taxonomy provides a wide range of objects (i.e. flows) in the form of material, signal and energy. Some terms on this taxonomy can be related to characteristics of an object rather than object per se, e.g. velocity. The description of Sickafus (1997) and Fey and Rivin (2005) pertain to tangible aspect of objects. Sickafus (1997) discusses that an object should possess at least the attributes of mass and volume. However, information and light are described as special cases by Sickafus (1997), that is, they can be described as an object at the designer`s discretion.

The SSFD adopts the concept of Dori (2002) in the definition of an object; it is given in Figure 3.4.

> *An object is a thing that has the potential of stable, unconditional physical or mental existence (Dori, 2002, p.57).*

Figure 3.4: OPM object definition adopted by the SSFD

As discussed earlier, the concept of Dori (2002) addresses both "tangible" and "intangible" aspects of an object throughout time. It covers all the points that have been raised by other researches. Furthermore, controversial terms in literature can also be addressed by Dori (2002)`s framework, e.g. light and information (see Sickafus (1997)).

3.1.2.1.2 Attributes in the Definition of a State

The statecharts of Harel (1987) and the implicit state representation of Dori (2002) directly show the value of a given element and an object, while Umeda et al. (1996) and the explicit state representation of Dori (2002) describe the value of an entity and an object through attributes. The term "attribute" is also used by Sickafus (1997) in the definition of an object. Goel et al. (2009) use the term "parameter" with the same meaning.

Umeda et al. (1990, p.182) quote attribute description of Tomiyama and Yoshikawa (1986): "a physical, chemical, mechanical, geometrical or other property which can be observed by scientific means." This description points out that an attribute should be measurable to be observed by scientific means. All approaches in Table 3.1 use the term "value" in the articulation of a measurable object attribute. According to Dori (2002, p.324), a value is "the concrete amount, quantity or specification of an attribute. Taken together, the SSFD describes "measurable" attribute of both facets (i.e. physical and mental existence) of an object based on the concepts of Umeda et al. (1990) and Dori (2002), which is given in Figure 3.5.

An attribute is a property which can be observed by scientific means (Umeda et al. (1990) and can be articulated in term of a concrete quantity (Dori (2002).

Figure 3.5: Attribute definition in the SSFD

As mentioned above, Sickafus (1997) suggested that a tangible object should possess at least the attributes of mass and volume, while Dori (2002) suggested that a physical object should have mass and occupy coordinates in space and time. Considering the controversial terms in literature (i.e. light), every object may not possess mass and volume. However, it could conceivably be hypothesised that space and time can be referred to as “global attributes” of an object, namely each object should possess these attributes. Flow-based approaches in Section 2.3.3 reflect the global attribute “time” by representing sub-functions of a system in respect of causality, while the FBS of Umeda et al. (1996) and the SBF of Goel et al. (2009) embed the time dimension into the function definition through behaviour which is described as sequential state transitions. The global attribute “space” is less emphasized in the reviewed function modelling approaches in Chapter 2.

The SSFD incorporates “time” and “space” as “global attributes” of an object as follows:

- The global attribute “time” is embedded into the state definition by representing transfer between states with respect to causality, i.e. an input state of a state transition is the output state of another state transition, and vice versa.
- The global attribute “space” is referred to the “location” of an object in which it is acted on and it is specified along with the object`s attributes (i.e. local attributes) in italic.

An updated version of state representation in Figure 3.3 is shown in Figure 3.6.

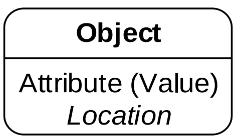
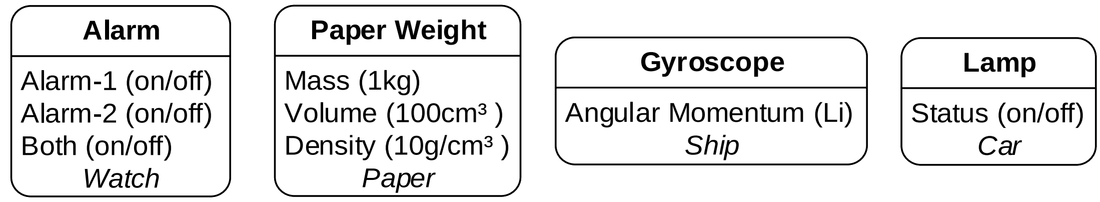


Figure 3.6: Schema of a state in the SSFD

State box in Figure 3.6 consists of two parts. The upper part contains the name of the object in bold and the below part contains the object attributes including local attribute(s) and global attribute “location”. Figure 3.7 shows state representation examples in Table 3.1 in terms of the SSFD state model.



a-Alarm of a Watch **b-Paper weight** **c-Gyroscope** **d-Lamp**

Figure 3.7: The use of the SSFD state model in the representation of state examples in Table 3.1

The object “alarm” in Figure 3.7-a is an intangible object and it is recorded on the watch which is its location. Sub-states of the alarm (e.g. alarm1) in Table 3.1 are shown as attributes in Figure 3.7-a with their possible values “on” and “off”. Unlike the FBS state representation of paper weight in Table 3.1, Figure 3.7-b represents the state of paper weight in a single representation. Angular momentum is shown as an attribute of the gyroscope in Figure 3.7-c. The difference to the SBF of Goel et al. (2009) is the clear separation between location attribute (i.e. ship) and local attribute (i.e. angular momentum) of the gyroscope. Figure 3.7-d represents the state of a lamp in a single representation.

The advantage of the SSFD state model over the approaches in Table 3.1 is the representation of a state in a single representation and in a structured way by differentiating between local attributes (e.g. size) and global attributes (i.e. location and time) of an object.

3.1.2.2 Function in the SSFD

Umeda et al. (1996, p.276) suggest that “function is an intuitive concept depending on the designer`s intention”. Table 2.1 in Section 2.4 supports this suggestion by representing different definitions of functions in the reviewed approaches. However, as noted by Vermaas (2013), the meaning of function can be generalized based on the design method used.

As discussed in Section 2.4, the most of the reviewed function modelling approaches define a function with respect to the flows of material, energy and information or state. Coherent with the proposed SSFD state model in the previous section, the OAF framework of Sickafus (1997) is adapted by the SSFD in the definition of a function; it is given in Figure 3.8.

An engineered function is defined in terms of the triad of an input state, an output state and a design solution which ensures the transfer between states by addressing relevant attribute(s) of the input state.

Figure 3.8: Function definition in the SSFD

The use of an arrow in the representation of a function is common practice in state-based approaches, e.g. the statecharts (Harel, 1987). Coherent with the principles of Harel (1987), an open arrow with the function text below is used to denote a function in the SSFD, as shown in Figure 3.9.

Figure 3.9: Function representation in the SSFD

The head of the arrow is located on the output state. The function text indicates that the attribute(s) of the output object is modified through combining relevant attribute(s) of the design solution with the attribute(s) of the input object, as illustrated in Figure 3.10. The design solution is represented in a grey box and it is thought of as an object described by a set of measurable attributes, like the SSFD state model.

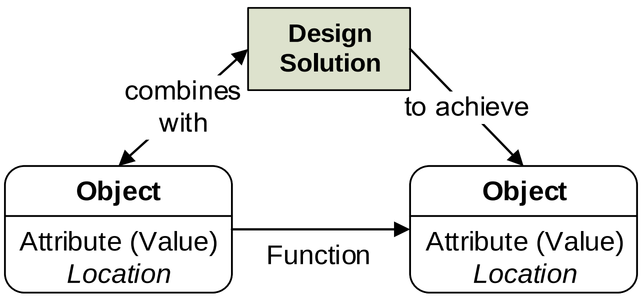
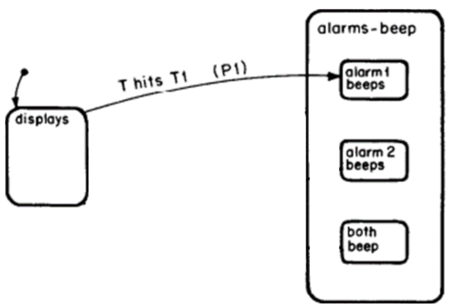
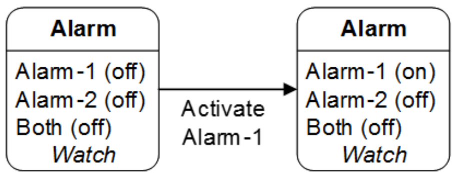
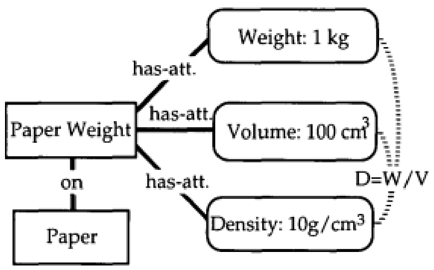
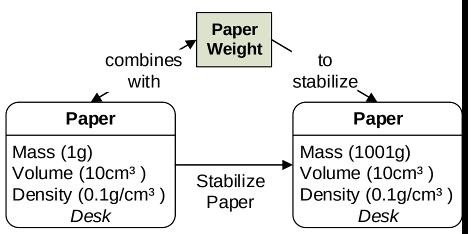


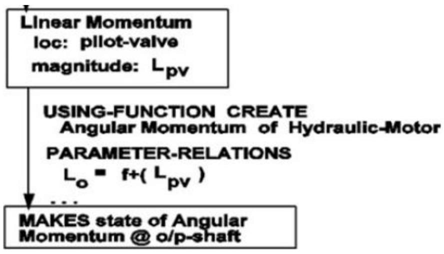
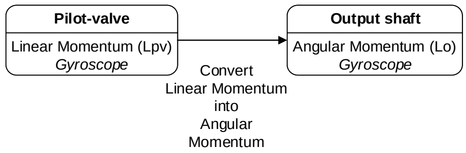
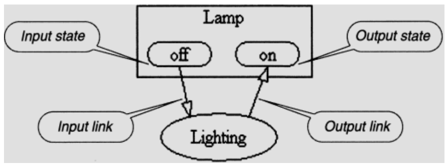
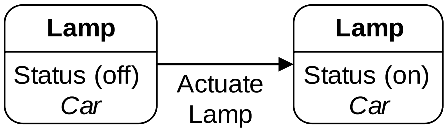
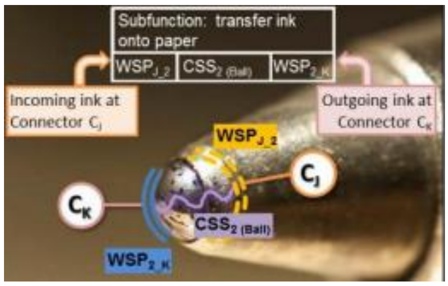
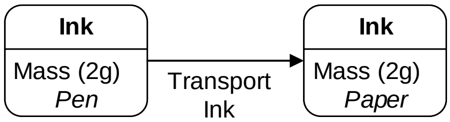
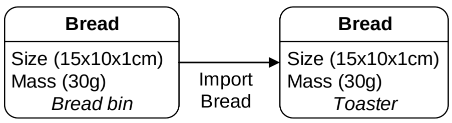
Figure 3.10: The triad of an input state, an output state and a design solution in the SSFD function definition

Section 2.4 showed that a function is commonly articulated in verb-noun format in respect of the flows of material, energy and information or state. Regarding the SSFD, this articulation is related to transfer between states. Considering the SSFD state model, the articulation of a function in verb-noun format is structured with respect to the OAF framework of Sickafus (1997). This characterization is related by the rule that the verb corresponds to the operation on the object attribute(s) and the noun to the object or the object attribute.

Section 3.1.2.1 compared the SSFD state model with the approaches which use states in function modelling. In this section, the SSFD function model is compared with these approaches as well as a flow-oriented approach (i.e. the FB of Stone and Wood (2000)) and a function-structure oriented approach (i.e. the C&C²-A of Matthiesen and Ruckpaul, 2012). Table 3.2 shows these approaches with an example and matching SSFD state transitions.

Table 3.2: Comparison of the SSFD function model with a selection of function modelling approaches reviewed in Chapter 2

Approach (Reference)	Example	SSFD state transition
Statecharts (Harel, 1987)	Alarm function of a Watch (based on p.237) **Function (event):** T hits T1 	
FBS model (Umeda et al., 1996)	Paper weight (p.276) **Function:** to keep paper from moving 	

SBF model (Goel et al., 2009)	A sub-function of a gyroscope (p.25) **Function:** Create angular momentum of hydraulic motor 	
OPM (Dori, 2002)	Lamp (p. 88) **Function sentence:** Lighting and Lamp function to ***make an object or area easy to see*** 	
C&C²-A (Albers and Zingel, 2011)	A sub-function of a ballpoint pen (p.4) **Function:** Transfer ink onto paper 	
FB (Stone and Wood, 2000)	A sub-function of a bread toaster (p.35) **Function:** import solid 	

The key points from Table 3.2 are summarized as follows:

- The example of the statechart shows the transition between the state “displays” and the sub-state “alarm1 beeps”. The respective internal time settings of the alarms are denoted by T1 and T2, while T shows the current time. The condition P1 abbreviates “alarm 1 enabled ^ (alarm 2 disabled v T1 ǂ T2).” The statechart articulates the event “T hits T1” in relation to the time setting “T1” of the alarm with the condition P1, as illustrated in Table 3.2. The state representation of the SSFD provides a compact representation by showing all beeping states of the alarm in terms of attribute values. The function is articulated with respect to the relevant attribute of the alarm, i.e. alarm-1. It should be noted the output state of the SSFD reflects the condition P1 by representing the output values of the attributes alarm-1 and alarm-2 as “on” and “off”, respectively. Sequential representation of the states refers to the time setting of the alarm, i.e. ΔT.

- With respect to the example of the FBS, the main advantages of the SSFD over the FBS are the representation of the state in a single box and the clear separation between function and design solution. The function “stabilize paper” is articulated and represented in a solution-neutral way.
- The SBF state transition in Table 3.2 describes the substances “linear momentum” and “angular momentum” with their parameters and parameter values. For the SSFD state model, the term “momentum” is an attribute of an object and the term “magnitude” is the value of the attribute “momentum”. The SBF state transition is converted into a SSFD state transition in Table 3.2, by convention, the parameter “location” is the object and the substance “momentum” is the attribute of the relevant objects in the SSFD, while the parameter “magnitude” is the attribute value. The function is articulated on the basis of the object attribute “momentum”.
- In terms of the example of the OPM, it seems that the SSFD state transition is providing more detailed and yet simpler representation of the function of the lamp by describing the lamp attributes accordingly. The function articulation is kept as succinct as possible compared to the OPL function sentence in Table 3.2.
- With regard to the example of the C&C^2-A, incoming ink and outgoing ink are shown at Connectors in Table 3.2. The WSPs and the CSS are described based on the elements of the pen, i.e. ball. While the example in Table 3.2 illustrates the function “transfer ink onto pen”, it is not clear how the function is articulated. The SSFD provides a solution-neutral representation by focusing on the flow of ink from pen to paper in terms

of a state transition and the function "transport ink" is articulated in respect of the global attribute "location" of the ink.

- As compared to the example of the FB, the SSFD provides a more detailed function representation in Table 3.2. The input and the output are described in terms of the bread state with measurable attributes including location and the function is articulated on the basis of the relevant bread attribute change (i.e. location) required to transfer between states, i.e. import bread instead of import solid.

Together these points on the comparison in Table 3.2 provide important insights into the concept of the SSFD in function modelling.

- The SSFD supports a compact function representation by using states and functions as the main elements.
- The functional representation is fully in the functional domain and solution-neutral. Referring to the axiomatic design of Suh (1990), the SSFD function model divorces the consideration of function from the consideration of the design solution by including the design solution conceptually in function modelling. Therefore, only the flow of state transitions through the system is represented without reference to a design solution.
- It is possible to hypothesise that thinking of a function in terms of a triad of an input state, an output state and a design solution facilitates solution-neutral definition of a function in respect of a state transition. For example, the SSFD models the example of the FBS in Table 3.2 in terms of the input and the output states of a paper instead of the paper weight. The function "stabilize paper" is articulated based on the paper attributes. A paper weight can fulfil this function as long as its relevant attribute (i.e. mass) combine with the mass of the paper (see corresponding SSFD in Table 3.2). Different design solutions can be used for the same state transition by following the same principle.

3.1.3 Representation of function chains based on the SSFD

Functional model of a system by the SSFD can be developed by defining all functions of the system on the basis of the triad defined in Section 3.1.2.2. The previous section showed the triad for one function and one state transition, as illustrated in Figure 3.10. This raises a question about using the SSFD function model in the development of a chain of functions.

As discussed in the beginning of Section 3.1, the demonstration of function modelling of an existing product based on reverse engineering is common practice in literature. Coherent with reverse engineering practice, the SSFD can develop functional model of an existing system based on the given set of functions of the system. However, this requires to follow a different way in the development of the model. As opposed to the proposed practice in Section 3.1.2.2, state transitions should be described based on the given functions. The states are connected to each other in a way that the output state from one function becomes the input to the next, and so on.

A function tree represents functions of a system hierarchically, showing the link between the main function and the sub-functions (Bertsche, 2008). Figure 3.11 shows function tree for an in-tank fuel delivery system (hereafter FDS) based Henshall and Campean (2009). Function tree in Figure 3.11 shows the FDS functions at three levels. The top level function is the main function. This follows the first level sub-functions and the second level sub-functions.

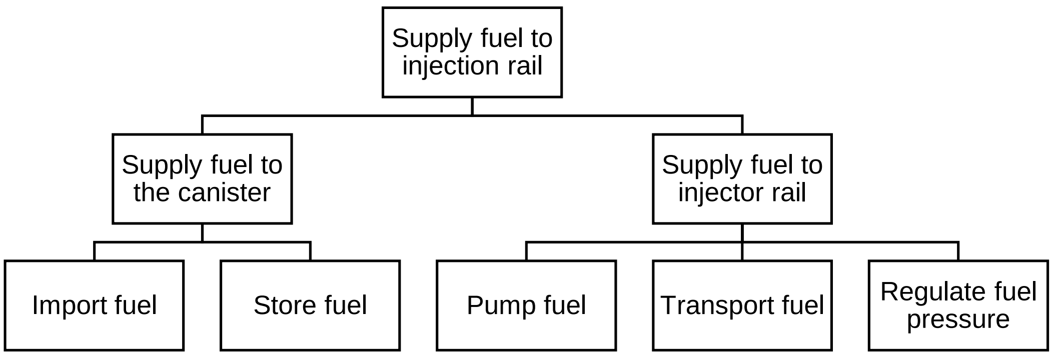


Figure 3.11: Function tree of an in-tank FDS (adapted from Henshall and Campean, 2009)

Figure 3.11 shows the functions at the second level with respect to causality from left to right. As discussed earlier, the SSFD can represent functional model of the FDS by identifying an input and an output state of each function and linking these states together. Figure 3.12 shows a FDS SSFD based on the given sub-functions in Figure 3.11. The box around the diagram shows the limits of the scope for the analysis.

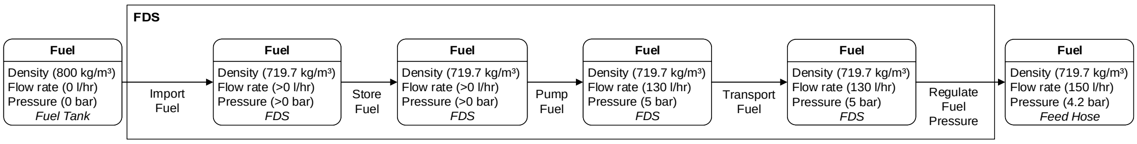


Figure 3.12: FDS SSFD

Figure 3.12 shows that the state of fuel through the FDS is represented in terms of the attributes “Density”, “Flow rate”, “Pressure” and “Location”. Each state transition in Figure 3.12 is described accordingly in respect of the corresponding function in Figure 3.11, which addresses the change of the relevant fuel attributes. The output state of a function should match the input state of the next function in Figure 3.12, and so on. For example, the function “import fuel” brings in the fuel from the fuel tank, as shown in Figure 3.12. As soon as the fuel in imported, it is stored within the FDS, as indicated by the function “store fuel”. It is therefore the output state of the function “import fuel” matches the input state of the function “store fuel” in Figure 3.12. Location of the fuel is referred to as “FDS” for all states within the SSFD, since design elements of the FDS have not been determined yet. Once they are determined, the SSFD in Figure 3.12 can be updated based on the selected design elements.

3.2 Review of research questions

As mentioned in the beginning of Section 3.1, an engineered system is commonly analysed on the basis of the flows of material, energy and information. The FDS SSFD in Figure 3.12 addresses the flow of material (i.e. fuel) by describing a sequence of state transitions with the aim of achieving the main function “supply fuel to injection rail” in Figure 3.11. While the FDS SSFD in Figure 3.12 focuses on the achievement of one main function requirement (i.e. mode of operation) based on one flow through the FDS, a complex system addresses multiple “main” functions and therefore include multiple flows of material, energy and information with respect to each main function. For example, Campean et al. (2011) described three main engineering function requirements for an electric vehicle powertrain. The analysis of Campean et al. (2011) shows that each function requirement requires different types of flows which are related to each other. In order to guide the use of the SSFD in function modelling of complex multidisciplinary systems, the following questions need to be addressed:

1) How do we capture all flows through a system using a SSFD?

2) How do we capture multiple modes of operation of a system in a single SSFD?

The next section describes research methodology followed to address these questions.

3.3 Research methodology for the development and the validation of System State Flow Diagram

The concept of Chandrasekaran and Josephson (2000) is used in the development of a SSFD in respect of environment- and device-centric view of a system. This concept also captures the points of the reviewed approaches in Chapter 2, e.g. Functional Basis model of Stone and Wood (2000) relates the overall function to the customer requirement (environment-centric view) and sub-functions represent the operation of the system in relation to the customer requirement (device-centric view).

The first research question described in the previous section is addressed by Chapter 4. The chapter refers to the environment-centric view, but focuses on the use of the SSFD function model in Figure 3.10 in the development of device-centric functional model of a system with one mode of operation (see Table 3.3).

Coherent with Suh (2005)`s complexity definition, it can be suggested that the more the flows of material, energy and information a system address, the more the functional model of the system gets complicated. This shows the need for a set of steps for the establishment of functional model of a system using the SSFD function model. The overview of the function modelling approaches in Section 2.3 shows that the process of developing functional model of a system involves extensive judgement of the practitioner, e.g. the description of sub-functions in the FAST. This means that there is always an element of subjectivity in function modelling of a system. The effect of this subjectivity on function modelling can be mitigated by providing explicit, prescriptive guidelines for the practitioner, which are termed “heuristics” by Maier and Rechtin (2010). Otto and Wood (2001) described the module heuristics (see Section 2.3.3.4) in the same manner. For the module heuristics of Otto and Wood (2001), functional model of a system consists of particular types of flows and functions. SSFD heuristics are introduced in Chapter 4 on the basis of the principles of the module heuristics for the development of functional model of a system using the SSFD function model in a structured way.

Chapter 4 focuses on theoretical development of the SSFD framework (i.e. SSFD function model and SSFD heuristics) for conducting function modelling of a system by testing and validating the proposed framework through application to a range of desktop case studies. Table 3.3 summarizes the plan for the case studies employed for the following chapters of the thesis.

Table 3.3: The case studies used in the following chapters of the thesis

Chapter		4				5		
Research Question		1				2		
Case Studies		Desktop Case Studies				Real World Case Studies		
		Framework Development	Framework Validation			Framework Development	Framework Validation	
		Bread Toaster	Glue Gun	Radiant Heater	Fuel Gauge	Front View Split Camera	Electric Vehicle Powertrain	Active Rear Spoiler
The basis of framework	State & Function	X	X	X	X	X	X	X
Flows through the system	Material	X	X	-	-	-	-	X
	Energy	X	X	X	X	X	X	X
	Information	X	X	X	X	X	X	X
View	Environment-centric	-	-	-	-	X	X	X
	Device-centric	X	X	X	X	X	X	X
Functionality (Mode of Operation)	One Function	X	X	X	X	X	-	-
	Multiple Functions	-	-	-	-	-	X	X

Both desktop and real world case studies address a combination of the flows of material, energy and information at system level, as shown in Table 3.3. The table also shows that the fundamental elements of the SSFD (i.e. state and function) are tested throughout Chapter 4 and Chapter 5.

As three separate heuristics which are incrementally applied for the establishment of solution independent functional model of a system are introduced in Chapter 4, a household bread toaster is used as an illustrative example since it supports the representation of all SSFD heuristics in a coherent way (see Section 4.1). The bread toaster is an electric device for making toast by applying heat to bread. This shows that the flow of material (i.e. bread/toast) is the “main” flow through the toaster which addresses the user requirement, i.e. producing a slice of toast.

The applicability of each proposed SSFD heuristic to the flows of material, energy and information is tested and validated on a glue gun, a radiant heater and a fuel gauge (see Section 4.2). These case studies address the “main” flows of material, energy and information, respectively. A glue gun provides the flow of a desired quantity of glue onto the required surface by melting a glue stick, while a radiant heater warms the environment by converting mains electricity into heat. A fuel gauge is a device that is used to make the amount of fuel contained in a tank known to the user. The flow of glue in the glue gun, the flow of heat in the heater and the flow of information (the amount of fuel) in the gauge are the “main” flows which are directly related to various customer requirements. By doing so, these case studies can also show how the SSFD heuristics support the establishment a link between a customer required function and functions of a device, which is one of the key weaknesses of the current SSFD, as discussed in Section 1.2. It is noteworthy that the working principles of the selected desktop case studies resemble other systems in real life, e.g. toaster and hairdryer, radiant heater and infrared lamp. Table 4.2 in Section 4.2 details the structure of the case studies.

The second research question described in the previous section is addressed by Chapter 5, as shown in Table 3.3. The SSFD heuristics represented in Chapter 4 aim to ensure that all flows through systems with one operation mode are captured using the SSFD function model. Chapter 5 focuses on further development and extension of the SSFD framework for function modelling of complex multidisciplinary systems with multiple operation modes in the same diagram. Case studies in this chapter address both environment- and device-centric function modelling of systems (see Table 3.3) and they possess two main characteristics;

- Complexity - they consist of multiple flows of material, energy and information considering different operation modes.
- Multidisciplinarity - they have features related to different engineering disciplines, e.g. mechanical, electrical, control, software.

Multidisciplinarity of a system can be determined in terms of the flows of material, energy and information, e.g. a control system includes mainly the flow of information.

As shown in Table 3.3, the developed SSFD framework are applied on real world (i.e. industrial) case studies for the test and the validation of practical applicability of the framework. A Front Split View Camera (FSVC) of a car makes objects external to the car (e.g. pedestrian) known to drivers via a small screen (i.e. driver interface) in the car. The FSVC has one mode of operation and it is reasonably a complex and multidisciplinary system, i.e. it includes control and electrical systems. Therefore, the concept of the developed SSFD framework is illustrated on the FSVC (see Section 5.3).

The applicability of the developed framework to complex and multidisciplinary systems with multiple operation modes is tested and validated on an electric vehicle powertrain (EVP) and an active rear spoiler (ARS) (see Section 5.4). The EVP addresses the flows of information and energy at system level, that is, it has features in respect of the disciplines of control and electricity. The ARS is an electro-mechanical system which consists of electronic, mechanical and control features embedded within the system.

The main difference between the EVP and the ARS is the way of addressing multiple operation modes. The EVP controls the flow of electrical energy through the vehicle to move the vehicle, to charge the vehicle and to power the vehicle accessories, like headlights and fans. The ARS controls the angle of spoiler in relation to the speed of the vehicle so as to manage the air flow around the vehicle. The EVP focus on the change of a local attribute (e.g. the voltage of electrical energy), while the ARS addresses the global attribute “location”, i.e. the angle of spoiler.

3.4 Chapter summary

This chapter started by introducing the development of the key concepts and elements of the SSFD, summarized as follows:

- Section 3.1.2.1 described a state in the SSFD as an object described by a set of measurable attributes including local attributes and global attributes of “time” and “location.

- Section 3.1.2.2 described a function in the SSFD as a triad of states. Design solution is thought of as a state in the SSFD. Its relevant attribute(s) join to the attribute(s) of the input state to generate the output state.

Section 3.1.3 explained the deployment of these elements to represent function chains of an existing system. The key research questions that arose from the findings from this application were phrased in Section 3.2. Section 3.3 introduced the research methodology to address these research questions.

4. Development of Heuristics for Function Analysis based on System State Flow Diagram

This chapter introduces the deployment of the SSFD function model presented in Chapter 3 to develop functional model of a system. The first part of the chapter describes and represents SSFD heuristics on a bread toaster. The validity of these heuristics to the flows of material, energy and information is tested on a range of desktop case studies in the second part of the chapter.

4.1 The method of SSFD heuristics for function modelling

In respect of the first research question in Section 3.2, this chapter seeks to explain how to ensure that all flows through a system are captured using the SSFD function model in a structured way.

As discussed in Section 3.3, function modelling approaches in literature require extensive judgement of the practitioner which affects “structured” development of functional models of systems. Maier and Rechtin (2010, p.55) described the term “heuristics” as “trusted, time-tested guidelines for serious problem solving”. In terms of the context of this research, coherent with Maier and Rechtin (2010), heuristics are referred to prescriptive guidelines on the development of function modelling of a system with the aim of mitigating the effect of subjectivity on the modelling activity caused by the practitioner, as mentioned in Section 3.3. Therefore, SSFD heuristics in this thesis can be described as a methodology of the deployment of the SSFD function model to develop functional model of a system on the basis of explicit, prescriptive guidelines. Three SSFD heuristics are incrementally applied as follows:

1) Main Flow Heuristic,
2) Connecting Flow Heuristic,
3) Branching Flow Heuristic.

As mentioned in Section 3.3, these heuristics are introduced and illustrated on a household bread toaster in the rest of Section 4.1. Considering an engineered system consists of the flows of material, energy and information, the validity of each SSFD heuristic to the flows of material, energy and information is tested on the desktop case studies “glue gun”, “radiant heater” and fuel gauge” in Section 4.2.

4.1.1 Main flow heuristic

4.1.1.1 Identification of the main flow

The main flow heuristic of a SSFD describes what a system is for – in other words, the purpose of a system. It is the first SSFD heuristic which is introduced in order to describe the first flow in the development of a SSFD. The principle underpinning this heuristic proposed in here is to focus on the intended effect of the system on the user. Stated succinctly, the main flow heuristic is described in Figure 4.1.

The main flow heuristic describes the purpose of a system by determining the flow related to the intended effect of the system on the user.

Figure 4.1: Description of the main flow heuristic

The approaches reviewed in Chapter 2 develop function model of a system at different levels of system design. For example, flow-oriented approaches in Section 2.3.3 differentiate between the high level function (i.e. overall function) and the lower level functions (i.e. sub-function) of a system and they represent what the system does by decomposing the high level function into lower level functions. Otto and Wood (2001) use the term “dominant flow” to refer to the flow through the set of sub-functions of a system, from entry of the flow in the system to exit from the system or conversion of the flow within the system, while Pahl et. al (2007) use the term “main flow” to represent the main functions of a system those directly address the fulfilment of its overall function.

According to Chandrasekaran and Josephson (2000), there are two views in function modelling of a system, i.e. device- and environment-centric. The effect of the device on the environment in which it operates (i.e. environment-centric view) is a result of the way of working of the device (i.e. device-centric view) related to the intended effect on the environment. As mentioned in Section 2.3.6.1, Deng (2002) describes the purpose of a design as a purpose function which is human-oriented and it describes the designer`s intention. The purpose functions are achieved by action functions which are human-related and they are used to describe intended behaviours of artefacts.

Like the flow-oriented approaches, a SSFD illustrates what a system does by representing its operation in terms of state transitions. However, coherent with the environment-centric view of Chandrasekaran and Josephson (2000), the diagram is developed with the aim of producing the intended effect on the environment in which the system operates. The environment may be the user or an object related to the intended effect on the user. Figure 4.2 shows the state of a user in terms of a SSFD state transition for the purpose of exemplifying the environment.

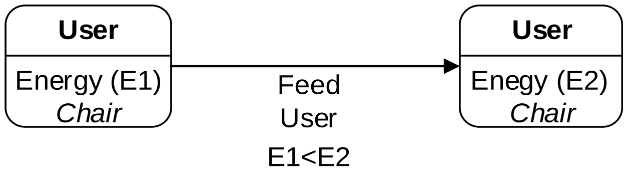


Figure 4.2: State transition for a user

Figure 4.2 shows the state of the user who is on a chair in terms of the attribute "energy" whose initial and final value are denoted by "E1" and "E2", respectively. The function "feed user" affects the user`s energy, as shown in Figure 4.2.

According to the SSFD function model in Section 3.1.2.2, a design solution is required for the transfer between states in Figure 4.2. A SSFD represents function model of the design solution and the main flow of the design solution is associated with the intended effect of the design solution on the environment, which is shown in terms of a state transition for the user in Figure 4.2. The output of the main flow addresses the achievement of the function "feed user" by combining with the input state of the user. Assuming that the user requires a slice of toast in respect of the function "feed user", the design solution can be named "toasting device". From a device centric perspective, the flow of bread, transformed into toast by the device can be considered as the "main flow" since the output of the main flow is directly associated with the intended effect of the device on the user, as shown in Figure 4.3.

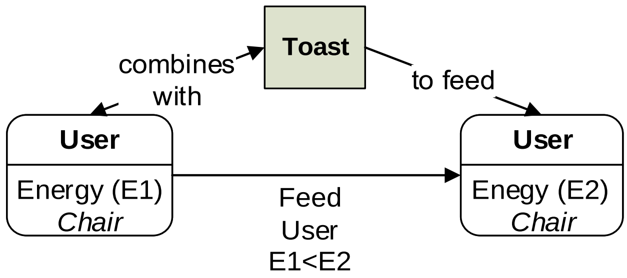


Figure 4.3: The triad of the input state “user”, the output state “user” and the output of the toasting device “toast” for the function “feed user”

On the basis of the SSFD state description formalism proposed in Section 3.1.2.1, the requirement of the user can be refined into specific and measurable requirements in terms of a state transition, which shows the input state and the output state of the main flow in the toasting device. Figure 4.4 shows a high-level SSFD for the main flow in the device. Grey box in Figure 4.3, “toast”, shows the design solution for the achievement of the function “feed user”. Like the SSFD state model, it is represented in terms of an object with measurable attributes, as shown as the output state in Figure 4.4.

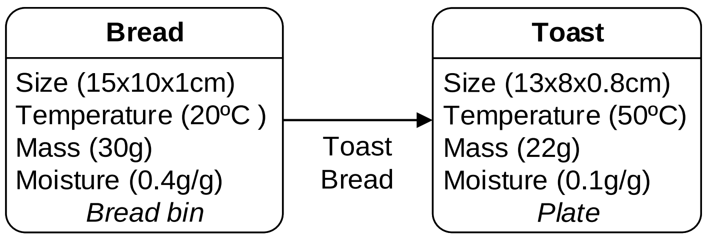


Figure 4.4: A high-level SSFD for the main flow in a toasting device

In fundamental terms the function “toast bread” in Figure 4.4 is achieved by heating the bread to 155ºC for a particular length of time, to allow chemical transformations (known as the Maillard reaction) in the bread to be triggered generating the characteristic flavours of toast (Mital et al., 2008). Further analysis of physics shows that heating the bread changes its length and thickness (i.e. size), temperature, mass and moisture holding capacity (hereafter moisture), as shown in Figure 4.4. Location of the bread (i.e. bread bin) and the toast (i.e. plate) are also specified during transition between states. The toasting device should address these key attributes of the bread for the generation of a slice of toast.

4.1.1.2 State-based Decomposition of the Main Flow

Coherent with the conversion-transmission module heuristic of Otto and Wood (2001), operation types on a state transition can be categorized into conversion and transmission. The former addresses the change in the attributes of an object that the function is applied to, while the latter is about changing the location of an object through the applied function. Figure 4.4 shows the conversion of bread into toast. The bread may undergo multiple conversions and transmissions for the generation of the toast. These operations on the state transition in Figure 4.4 are addressed by describing state transitions between the states in Figure 4.4. The identification of these state transitions requires the description of the way of achievement (see Kitamura and Mizoguchi, 2003) of the state transition in Figure 4.4, that is, physical phenomena that regulate the changes of the state attributes. As noted by Umeda et al. (1996), we can reason out intermediate states between the input state and the output state in Figure 4.4 from the input state based on the described physical phenomena. This supports the decomposition of the state transition in Figure 4.4 in an objective way by representing the behaviour of the state from the input state. Umeda et al. (1996) described this as “behaviour-state (B-S) relationships” in Section 2.3.5.2. The same section pointed out that function behaviour (F-B) relationships are subjective since the intentions of the users may be different, e.g. using the toasting device for heating baguette (French stick). Like the input state and the output state, an intermediate state is thought of as an object with its measurable attributes. Once intermediate states are identified, the flow of these states is mapped between the input state and the output state with respect to causality. The last step is to articulate functions required to achieve these state transitions. The same principles apply to the development of all state flow diagrams.

The way of achievement in the process of toasting bread is associated with the increase of the bread temperature. There are three basic ways of heat transfer: convection, radiation and conduction (Santanam et al., 1997) and there are a variety of ways of increasing the temperature of the bread on the basis of these modes, e,g. over an open fire (convection), by radiant heat (radiation) and on a grill pan (conduction), as shown in Figure 4.5.

Convection **Radiation** **Conduction**

Figure 4.5: Increasing the bread temperature by means of heat transfer

Solution neutral decomposition of the state transition in Figure 4.4 gives a process of toasting bread under any type of heat transfer represented in Figure 4.5. Figure 4.6 shows SSFD for the main flow in a generic solution independent toasting device.

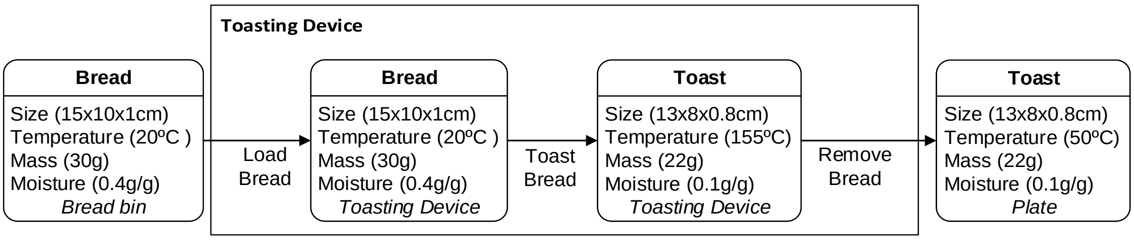


Figure 4.6: The main flow through the toasting device

It is assumed that the bread is located in bread bin and therefore the flow starts by bringing in the bread from outside the system boundary (i.e. load bread). Once the bread is toasted (i.e. toast bread) in the device, it is sent outside the system boundary (i.e. remove toast), e.g. plate. The salient points in Figure 4.6 can be highlighted as follows:

- The functions “load bread” and “remove toast” in Figure 4.6 address the change of the bread location and the toast location, respectively.
- The function “toast bread” in Figure 4.4 addresses all attributes of the bread including global attributes “location” and “time”. It shows the conversion of a slice of bread in bread bin into a toast on plate. The function “toast bread” in Figure 4.6 is about the process of toasting by applying heat to bread which is retained in the device.
- The toast emits thermal radiation as it cools. The function “remove toast” reflects this by representing a reduction in the toast temperature.

- Heating the bread to 155°C affects all key attributes of the bread, as discussed before. Time passes during the function “toast bread” reflects the time of exposure for the bread to heat for the generation of the output state in Figure 4.6. Time is implicitly shown in Figure 4.6 by specifying object attributes with respect to time and it is addressed by introducing a connecting flow (see the next section). If the timing is excessive, the toaster burns the toast.

4.1.2 Connecting flow heuristic

Figure 4.6 shows the conversion and the transmission operations between the input state and the output state of the toasting device main flow. In order to achieve the conversion operations, the flows of additional resources need to be connected to the conversion operations on the main flow. This heuristic aims to identify these flows and it is therefore termed Connecting Flow Heuristic. The connecting flow heuristic is stated formally in Figure 4.7.

> *The connecting flow heuristic aims to address the fulfilment of the conversion operations on the main flow by determining the flows of additional sources and connecting these additional flows to the conversion operations.*

Figure 4.7: Description of the connecting flow heuristic

The toasting device requires the connecting flows of energy and information.

4.1.2.1 The flow of energy

For the toasting device, the function “toast bread” in Figure 4.6 changes the composition of the bread (i.e. sugars and starches start to caramelize) and all key attributes of the bread are modified. As discussed in Section 4.1.1, the bread can be heated by any type of heat transfer. For engineering illustration purpose, the design solution is assumed to be a common household bread toaster which uses thermal radiation (radiant heat) to heat the bread will be considered (i.e. the design team choice for this solution based on evaluation of customer needs) in this section. Based on this assumption, the toaster must apply thermal radiation directly to the bread slice (Mital et al, 2008). According to the SSFD function model, the radiation combines with the bread to toast the bread, as shown in Figure 4.8.

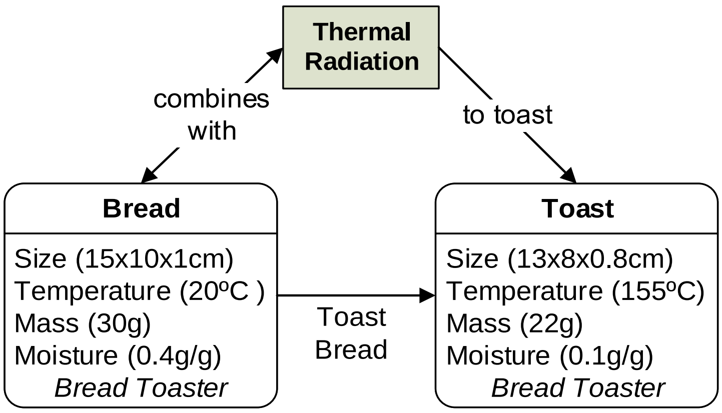


Figure 4.8: The triad of “bread”, “toast” and “thermal radiation” for the function “toast bread”

Figure 4.8 shows that the connecting flow should produce thermal radiation for the fulfilment of the function “toast bread”. A variety of sources of energy (e.g. electrical) and a range of design concepts (e.g. Nichrome wire) can be considered as ways of producing thermal radiation. A common household bread toaster uses electrical energy (EE) of mains electricity supply as energy source, which is converted into thermal radiation. Figure 4.9 shows this conversion operation in terms of a state transition.

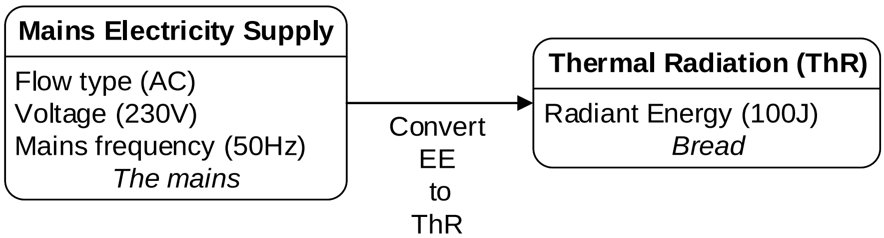


Figure 4.9: A high-level SSFD for the flow of energy in the toaster

Mains electricity supply is a general-purpose alternating-current (AC) electric energy supply and it is characterized by flow type, voltage and frequency in Figure 4.9. Thermal radiation consists of electromagnetic waves which can carry radiant energy to object with which they interact.

The input state and the output state of the energy flow in Figure 4.9 can be decomposed based on the same principles followed in the decomposition of the main flow of the toaster, i.e. by focusing on the way of achievement in the state transition. Figure 4.10 represents the flow of energy through the toaster.

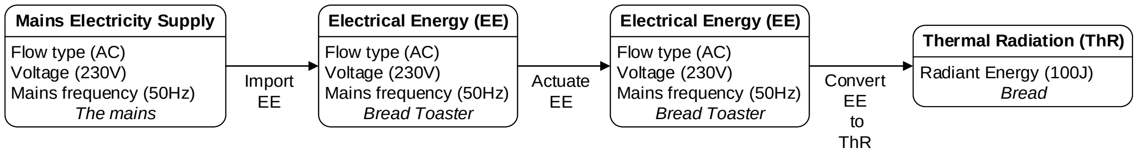


Figure 4.10: The flow of energy through the toaster

Reflecting the energy source choice made (i.e. Mains Electricity Supply) – the flow starts by bringing in electrical energy from outside the system boundary (i.e. import EE). Once the flow of electrical energy commences (i.e. actuate EE), it is converted into thermal energy (i.e. convert EE to ThR). Both input and output locations of the electrical energy for the function "actuate EE" are shown as "Bread Toaster", showing that the energy flows through the toaster. Location attributes of the electrical energy can be specified once relevant design elements are identified, e.g. the input location of the electrical energy for the function "actuate EE" may be "cable" assuming that the output location of the electrical energy for the function "import EE" is "cable".

It is noteworthy that while "energy" is shown in Figure 4.2 as an attribute of the user, "electrical energy" in Figure 4.10 is shown as an object. Electrical energy is not tangible. However, some instrument can prove their existence which can be described through measurable attributes, e.g. flow, frequency. The identification of thermal radiation as an object is based on the same principle. This shows that not every energy can be described as an object. For example, user is considered as a source of energy in Figure 4.2, hence, energy is shown as an attribute of the user.

The flow of energy in Figure 4.10 can be linked to the main flow in Figure 4.6 by linking the output state "thermal radiation" of the energy flow to the input state of the function "toast bread" on the main flow. Figure 4.11 shows an updated SSFD which includes the flow of energy through the toaster.

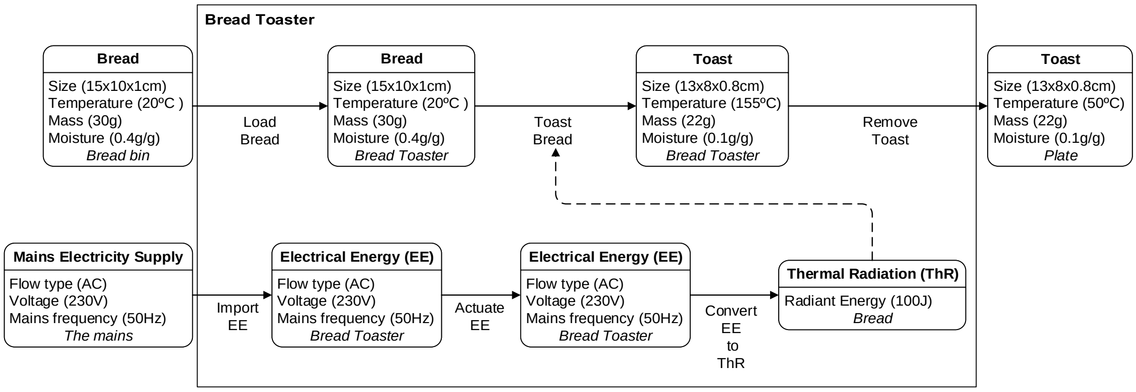


Figure 4.11: SSFD for the toaster: the flow of energy

It has been discussed in Section 2.3.3 that flow-oriented approaches represent a function in a box and link two functions by using an arrow, denoting the flow of material, energy and information/signal. Similarly, the FAST (Kaufman and Woodhead, 2006) and the IDEF0 (Buede, 2009) use a box in the representation of a function and they use a line and an arrow to link functions, respectively. The statecharts (Harel, 1987) use arrows to link states.

For the SSFD function model, the energy flow in Figure 4.11 explains how the conversion function “toast bread” on the main flow is achieved. The output state of the energy flow “thermal radiation” behaves as a design element for the achievement of the function “toast bread” by combining with the bread. Figure 4.8 showed this link using two types of arrows for the purpose of clarifying the triad, but in practice this link is shown by one arrow in the SSFD. The main difference to the current representations in literature is that the arrow from the output state is pointed to the function text related to the conversion to highlight the triad. This arrow is represented with a dashed line to distinguish it from the function arrow, as shown in Figure 4.11. Having specified the attributes of the output state “thermal radiation” of the energy flow, the triad of the input state (i.e. bread), the output state (i.e. toast) and the design solution (i.e. Thermal Radiation) in Figure 4.11 can be articulated as “Radiant Energy of Thermal Radiation combines with Bread of Toaster to toast Bread”.

4.1.2.2 The flow of information

The energy flow in Figure 4.11 shows how to toast bread. The conversion function “toast bread” is a time-dependent function. As discussed in Section 4.1.1, the toaster may generate a burnt toast as a result of over-toasting. This requires controlling the flow of energy through the toaster in respect of the function “toast bread” on the main flow. The function “actuate EE” in Figure 4.11 commences the flow of electrical energy in response to a control signal. Therefore, the design choice for the control of the time of exposure for the bread to the thermal radiation can be made based on the function “actuate EE”. This requires the generation of a control signal from a given input source. The signal combines with the input state of the function “actuate EE”, as shown in Figure 4.12.

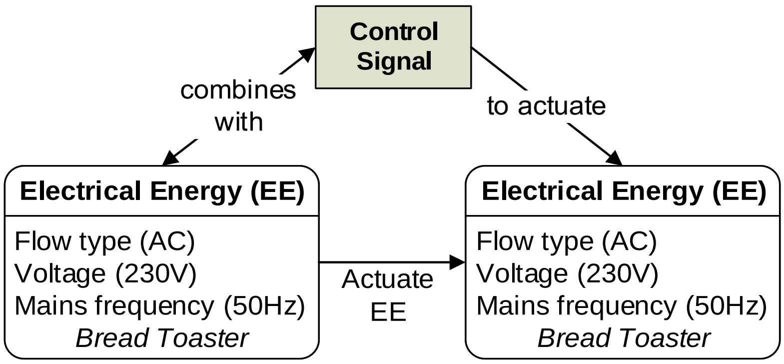


Figure 4.12: The triad of “input EE”, “output EE” and “control signal” for the function “actuate EE”

A toasting device can control the time of exposure for the bread to the thermal radiation by providing manual control which is adjusted by the user or automated control based on process measurements, namely by sensing a specific attribute of the toast, e.g. browning level. For a common household bread toaster, the control signal can be generated based on a manual type of control of the toaster. This requires the conversion of the user input (e.g. energy) into a control signal, as shown in Figure 4.13.

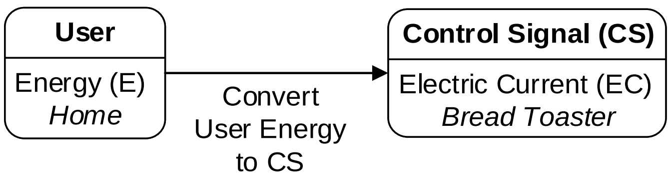


Figure 4.13: A high-level SSFD for the flow of information in the toaster

Reflecting the design choice made, Figure 4.13 shows that the toaster conveys the user`s intention (e.g. switching the toaster on/off) by converting the user energy to a control signal. The value of the user energy and the control signal`s electric current are denoted by “E” and “EC”, respectively. The converted signal is passed to relevant state on the flow of energy. Therefore, the conversion function in Figure 4.13 should be followed by a transmission function, as indicated in Figure 4.14.

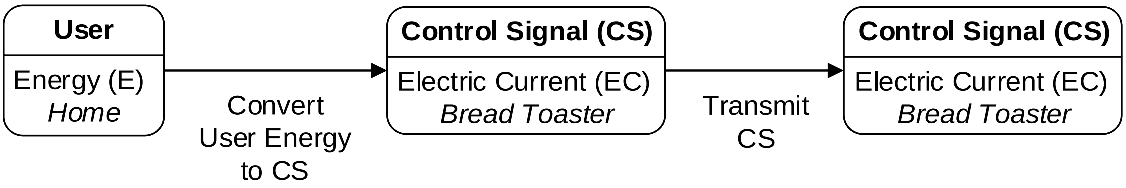


Figure 4.14: The flow of information through the toaster

The flow of information in Figure 4.14 can be linked to the energy flow of the toaster SSFD in Figure 4.11 by linking the output state “control signal” to the input state of the function “actuate EE”, represented as a triad in Figure 4.12. Figure 4.15 shows an updated SSFD which includes the flow of information through the toaster.

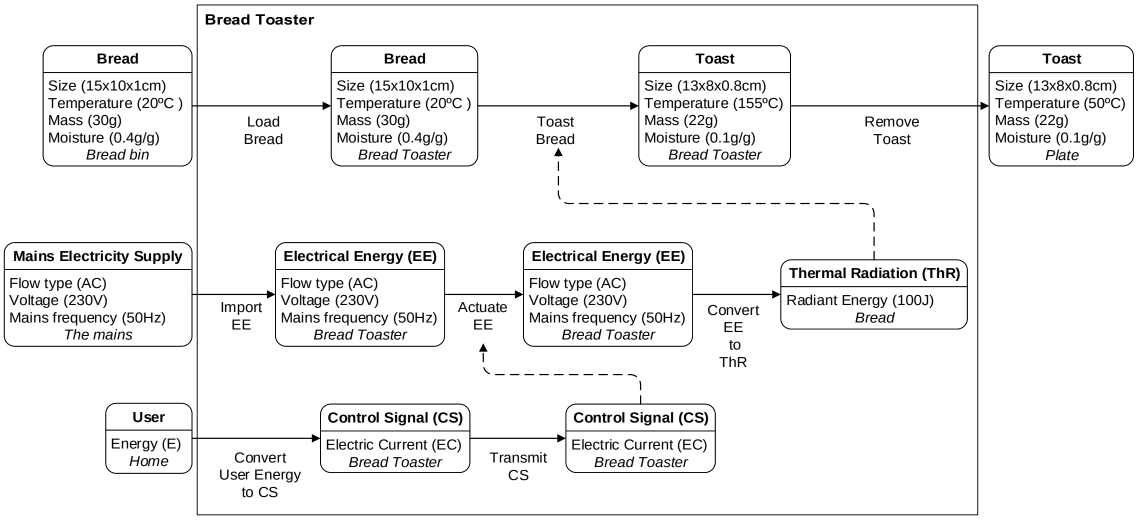


Figure 4.15: SSFD for the toaster: the flow of information

Similar to the connection of the energy flow to the main flow in the previous section, the triad in Figure 4.12 is indicated by a dashed arrow from the control signal to the function “actuate EE” in Figure 4.15.

4.1.3 Branching flow heuristic

An object attribute can be modified through other attributes of the same object. For example, the function “toast bread” in Figure 4.15 increases the bread temperature which affects other attributes of the bread, i.e. size, mass and moisture. The flow of states outside the toaster can be described for each object attribute change. Branching Flow Heuristic aims to describe this type of flows and it is succinctly stated in Figure 4.16.

The branching flow heuristic describes flow(s) that branches out of the main flow by focusing on each object attribute change on the main flow.

Figure 4.16: Description of the branching flow heuristic

As mentioned above, the function “toast bread” modifies the attributes “Size”, “Mass” and “Moisture” of the bread. The key points regarding these modifications can be summarized as follows:

- Increasing the temperature of a slice of bread decreases the bread moisture which affects the size and the mass of the bread, as shown in Figure 4.15. This shows that it is not necessary to represent branching flows for a decrease in the bread size and the bread mass, since they are modified due to vaporization of the bread moisture and changes in the bread composition as a result of an increase in the bread temperature.

- Toasting bread removes moisture from the bread in the form of vapour which is released into the atmosphere through evaporation. The toaster should direct the course of vapour outside the system boundary, as shown in Figure 4.17. The value of the vapour mass and the vapour temperature are denoted by “M” and “T”, respectively.

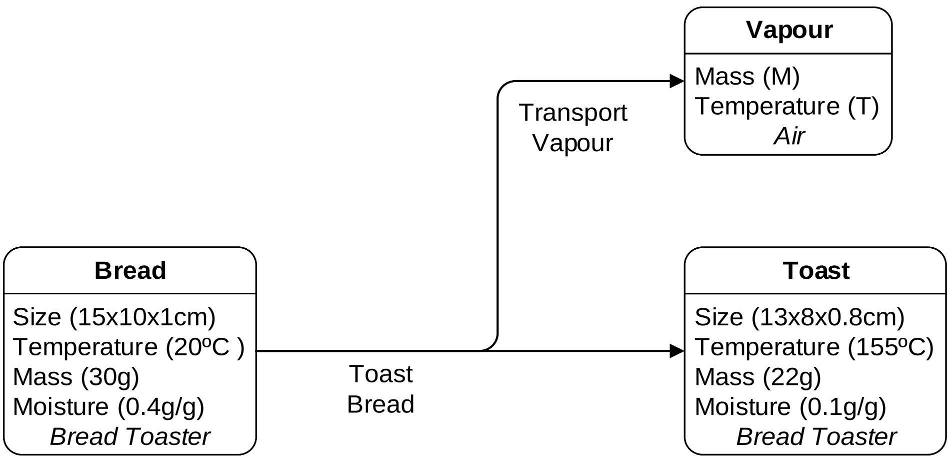


Figure 4.17: The flow of vapour through the toaster

In the case of the function “remove toast” in Figure 4.15, the toast with a high temperature emits thermal radiation. It is assumed that this radiation heats the air in the environment for the purpose of practicality. However, this heat transfer is assumed to be negligible and it is not necessarily shown on the toaster SSFD. On the basis of these findings, Figure 4.18 shows an updated SSFD which includes the flow of vapour branching out of the main flow.

Coherent with the branching flow heuristic of Otto and Wood (2001), Figure 4.18 shows that the flow of vapour must interface with the input state of the function “toast bread” on the main flow. The flow of vapour is branched out of the main flow after the function text “toast bread” since the vapour is generated due to this function. It is important to note that the flow of material and energy obey the “laws of conservation”, that is, they are conserved during the transition of states in the SSFD.

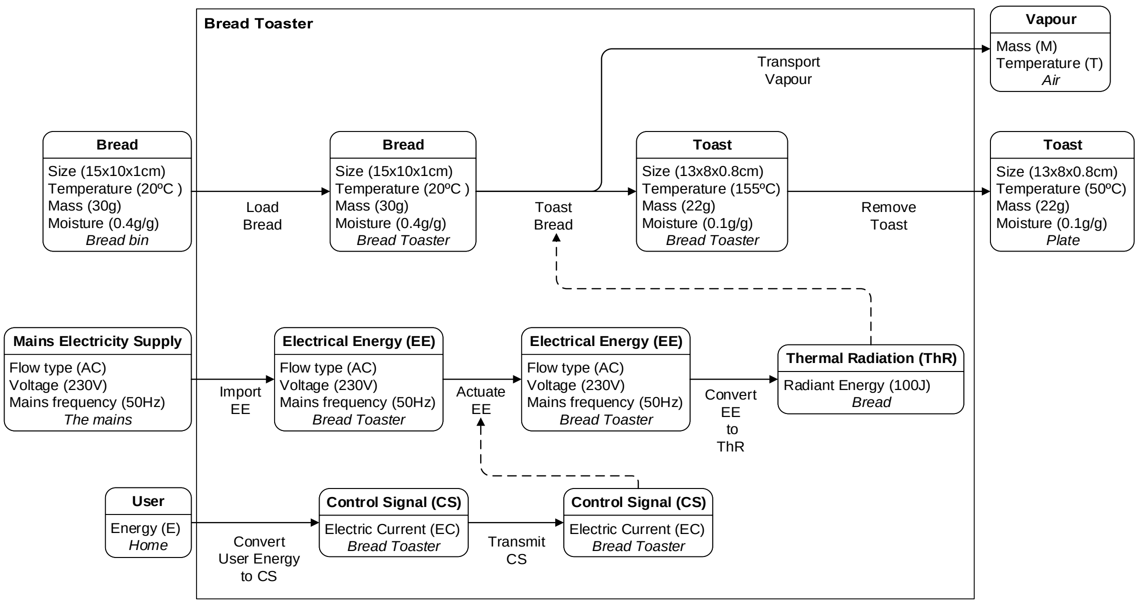


Figure 4.18: SSFD for the toaster: the flow of vapour branching out of the main flow

Table 4.1 shows object attribute relations in terms of an equation for each function of the toaster SSFD in Figure 4.18.

Table 4.1: Object attribute relations in the toaster SSFD

Function	Object attribute relations
Load Bread	Bread(15x10x1cm,20ºC,30g,0.4g/g,Bread Toaster) = f(Bread (15x10x1cm,20ºC,30g,0.4g/g,Bread Bin))
Toast Bread	Toast(13x8x0.8cm,155ºC,22g,0.1g/g,Bread Toaster) = f(Bread (15x10x1cm,20ºC,30g,0.4g/g,Bread Toaster).Thermal Radiation(100J,Bread))
Remove Toast	Toast(13x8x0.8cm,50ºC,22g,0.1g/g,Plate) = f(Toast (13x8x0.8cm,155ºC,22g,0.1g/g,Bread Toaster)
Import EE	Electrical Energy(AC,230V,50Hz,Bread Toaster) = f(Mains Electricity Supply (AC,230V,50Hz,The mains))
Actuate EE	Electrical Energy(AC,230V,50Hz,Bread Toaster) = f(Electrical Energy (AC,230V,50Hz,Bread Toaster))
Convert EE to ThR	Thermal Radiation(100J,Bread) = f(Electrical Energy (AC,230V,50Hz, Bread Toaster))
Convert User Energy to CS	Control Signal(EC,Bread Toaster) = f(User (E,Home))
Transmit CS	Control Signal(EC,Bread Toaster) = f(Control Signal (EC,Bread Toaster))
Transport Vapour	Vapour (M,T,Air) = f(Toast(13x8x0.8cm,155ºC,22g,0.1g/g,Bread Toaster)- Bread (15x10x1cm,20ºC,30g,0.4g/g,Bread Toaster)

Mathematical correctness of material and energy conservation in the SSFD can be shown by the equations in Table 4.1, provided all object attribute values are known. As indicated by the equation of the function “toast bread” in Table 4.1, it is assumed that all thermal radiation is transmitted to the bread. In practice, it will go part in the bread and the rest goes somewhere else (e.g. the atmosphere with the vapour) during the process of toasting bread. Identification of this branching flow requires the analysis of the flow of energy through the toaster as the main flow in its own right (see Section 4.2.2 for an example).

4.2 Application Examples

Section 4.1 represented the SSFD heuristics on a household bread toaster whose main flow is material. As discussed in Section 3.2, functional model of a system may consist of the flows of material, energy and information. This section aims to test the validity of each SSFD heuristic to the flows of energy and information, as well as material. While the case studies were selected based on the flow type of the main flow heuristic, they also enable to test of the connecting and the branching flow heuristics. Table 4.2 summarizes the structure of the case studies including the bread toaster which was analysed in the previous section.

Table 4.2: The structure of the case studies

Criteria		Bread Toaster	Glue Gun	Radiant Heater	Fuel Gauge
SSFD Heuristics	Main Flow (**Type:** Objects)	**Material:** Bread, Toast	**Material:** Glue	**Energy:** Mains Electricity, Thermal Radiation	**Information:** Control Signal
	Connecting Flow (**Type:** Objects)	**Energy:** Mains Electricity, Thermal Radiation, User (Energy)	**Energy:** Mains Electricity, Thermal Radiation, User (Energy)	**Energy:** User (Energy)	**Energy:** Electrical Energy
		Information: Control Signal	**Information:** Control Signal	**Information:** Control Signal	**Information:** Control Signal
	Branching Flow (**Type:** Objects)	**Material:** Vapour	**Energy:** Thermal Radiation	**Energy:** Thermal Radiation	-
Intended Effect on the Environment (Environment-centric View)		Feeding User	Closing Box	Warming User/Object	Making the contents of the fuel tank known to the driver
Main Function (Device-centric View)		Toast Bread	Melt Glue	Convert Mains Electricity to Thermal Radiation	Display the level of fuel

The salient points in Table 4.2 are summarized as follows:

- The row "SSFD heuristics" shows which heuristic addresses what flow and represents object(s) on the relevant flow. Table 4.2 shows that the heuristics on the case studies address a combination of the flows of material, energy and information. Branching flow heuristic represented in this chapter aimed to identify unintended by-products of the case studies with a focus on their main flows. While the flow of material (i.e. vapour) in the toaster SSFD and the flow of energy (i.e. thermal radiation) in the glue gun and the radiant heater SSFDs are shown as branching flows out of their main flows, none of the case studies in this chapter including the fuel gauge does not represent the flow of information as a branching flow (as indicated in Table 4.2) because the flow of information is not an unintended by-product.
- The row "intended effect on the environment" reflects environment-centric view of the SSFD by showing what the case studies are for. As shown in Table 4.2, the main flows of bread toaster and glue gun are material, however their intended effects on the environment are different. While the toaster changes the state of the user by producing a slice of

toast, the glue gun focuses on the state of an object (e.g. box) in respect of a user requirement (e.g. the need for closing a box).

- The row “main function” reflects the development of the case study SSFDs from a device-centric perspective by representing the high-level functions of the main flow SSFDs of the case studies.

The following sections represent the deployment of the SSFD heuristics to develop function model of the case studies “glue gun”, “radiant heater” and “fuel gauge”.

4.2.1 Glue Gun

As mentioned in the beginning of this section, the main flow heuristics of the bread toaster and the glue gun case studies address the flow of material. The main flow of the toaster is directly related to the utility to the user (see Figure 4.3). The glue gun case study in this section focuses on the representation of a SSFD whose main flow is indirectly related to the user.

A glue gun has a variety of uses including closing a box and assembling a toy. For any use, the gun should provide the flow of a desired quantity of glue onto the required surface. For example, in the case of the need for closing a box, the decision of the user is assumed to be using a glue gun out of a number of alternatives (e.g. sticky tape). This requires the gun to generate glue which is the output of the glue gun main flow. The glue combines with the required part of the box, as shown in Figure 4.19.

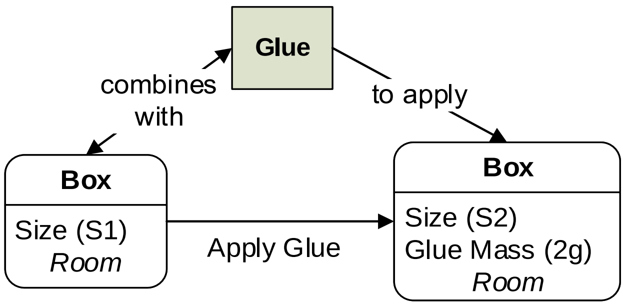


Figure 4.19: The triad of the input state “box”, the output state “box” and the output of the glue gun main flow “glue” for the function “apply glue”

Figure 4.19 shows the input and the output states of the box in a room in terms of its size value “S1” (open) and “S2” (closed). The mass of the applied glue is shown on the output state of the box. The glue in Figure 4.19 should be produced from a given input source. For a glue gun, the gun melts part of a glue stick and directs the melted glue onto the box surface. Figure 4.20 shows a high-level SSFD for the main flow in a glue gun.

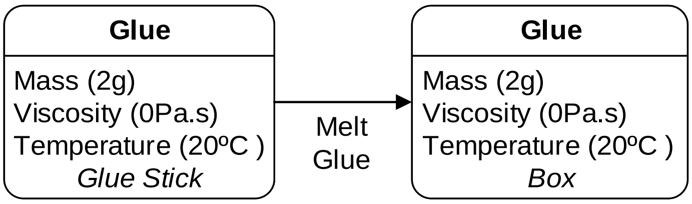


Figure 4.20: A high-level SSFD for the main flow in a glue gun

As shown in Figure 4.20, once the melted glue solidifies, it is assumed that it possesses the same attribute values (i.e. mass, viscosity and temperature) of the solid glue. For the purpose of engineering illustration, a household high temperature glue gun, which melts a glue stick by heating it to 190ºC, is analysed in this case study. Working principle of any glue gun is based on the fundamental working principle that the output state “glue” in Figure 4.20 is generated by melting the input state “glue stick” via a heat source. Conversion and transmission operations during state transition in Figure 4.20 can be described by decomposing this state transition based on the working principle of the gun. Figure 4.21 shows SSFD for the main flow in a glue gun.

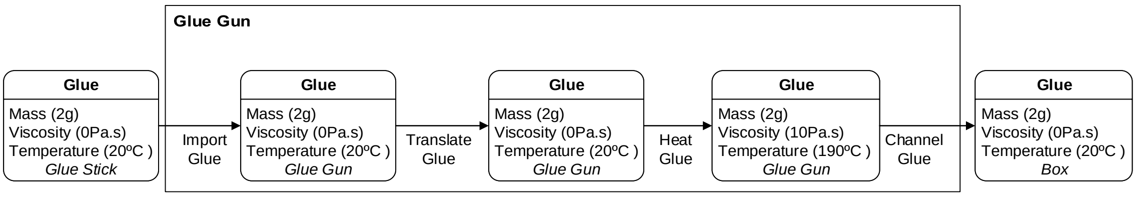


Figure 4.21: The main flow through a glue gun

Figure 4.21 shows conversion and transmission activities in a high temperature glue gun based on its fundamental working principle. The flow through the gun starts by bringing in the solid glue (i.e. glue stick) from outside the system boundary (i.e. import glue). This follows the movement of the glue into a linear direction (i.e. translate glue). As soon as the glue is melted (i.e. heat glue), the melted glue is directed onto the box surface (i.e. channel glue), as shown in Figure 4.21.

The function “heat glue” is the only conversion function on the main flow of the glue gun in Figure 4.21. It aims to melt the glue and this is shown in Figure 4.21

in terms of the output glue attributes “viscosity” and “temperature”. A variety of sources of energy can be considered as a way of heating the glue. Like the bread toaster case study in Section 4.1, thermal radiation can be applied to the glue. This shows that the connecting flow in respect of the function “heat glue” should generate thermal radiation which combines with the glue to heat glue, as shown in Figure 4.22.

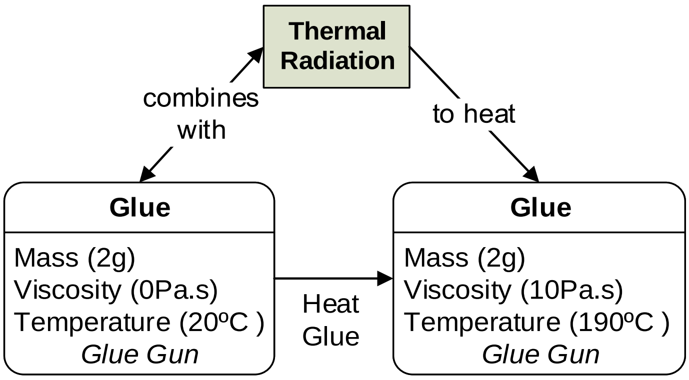


Figure 4.22: The triad of the input state “glue”, the output state “glue” and the output of the glue gun connecting flow “thermal radiation” for the function “heat glue”

Energy source for thermal radiation generation can be mains electricity supply. Figure 4.23 shows the conversion of electrical energy (EE) of mains electricity supply into thermal radiation in terms of a state transition.

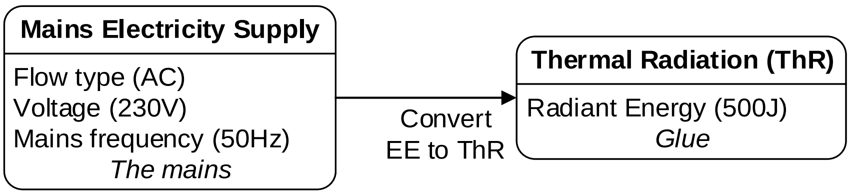


Figure 4.23: A high-level SSFD for the flow of energy in the glue gun

Figure 4.24 shows the decomposition of this state transition into intermediate state transitions in respect of the way of achievement of the state transition.

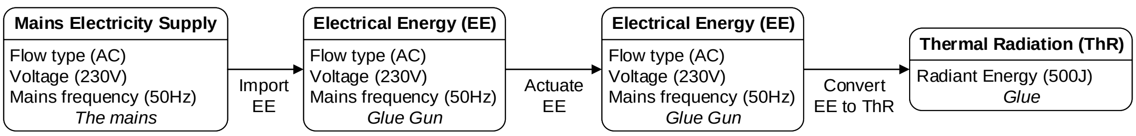


Figure 4.24: The flow of energy through the glue gun

The output state of the energy flow in Figure 4.24 is connected to the input state of the function “heat glue” in Figure 4.21, showing the achievement of the function “heat glue”. Figure 4.25 shows an updated glue gun SSFD which includes the flow of energy through the gun.

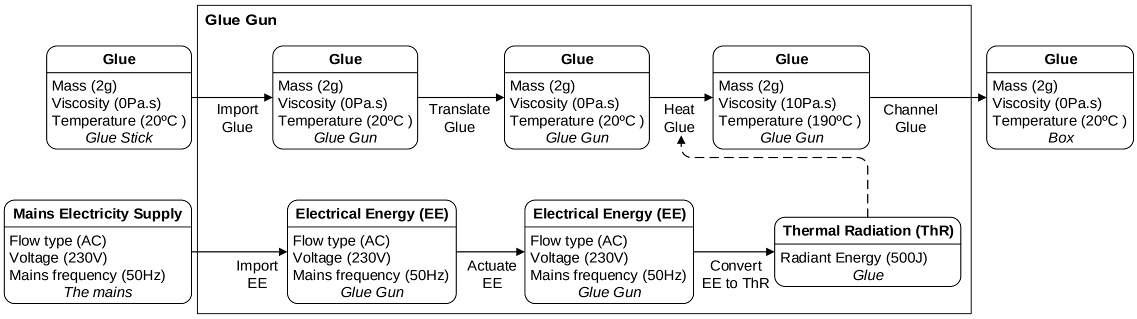


Figure 4.25: SSFD for the glue gun: the flow of energy

Time required to melt the glue depends on the duration of the process of heating glue using thermal radiation. The design choice for the control of the time of exposure for the glue stick to the thermal radiation can be made based on the function “actuate EE” which commences the flow of electrical energy in response to a control signal.

In a household glue gun, the user controls the operation of the gun manually. Therefore, the control signal can be generated based on the user input (i.e. energy) and it is transmitted to the input state of the function “actuate EE”, as shown in Figure 4.26.

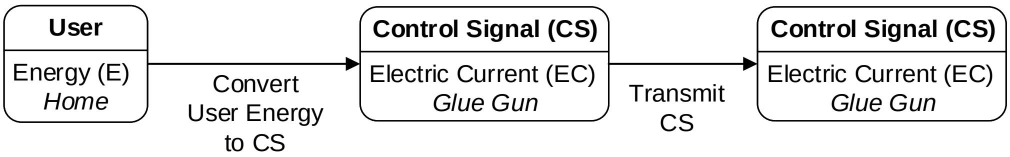


Figure 4.26: The flow of information through the glue gun

The value of the user energy and the control signal`s electric current are denoted by “E” and “EC”, respectively. The output state of the information flow in Figure 4.26 combines with the input state of the function “actuate EE” to perform the function “actuate EE”. Figure 4.27 shows an updated glue gun SSFD which includes the flow of information through the glue gun.

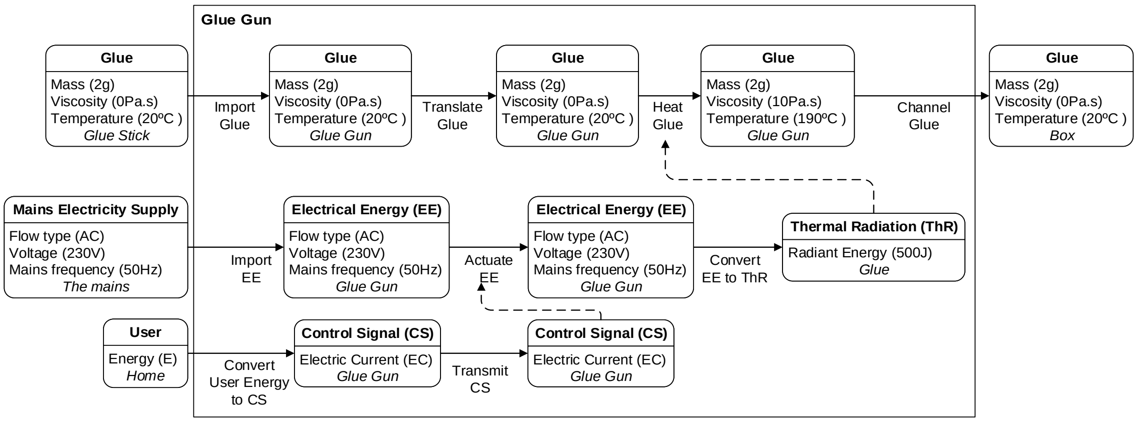


Figure 4.27: SSFD for the glue gun: the flow of information

In the case of the function “channel glue”, the glue with a high temperature in the glue gun emits thermal radiation. The gun should direct the flow of thermal radiation outside the system boundary, as shown in Figure 4.28 in which the value of the radiant energy is denoted by “RE”.

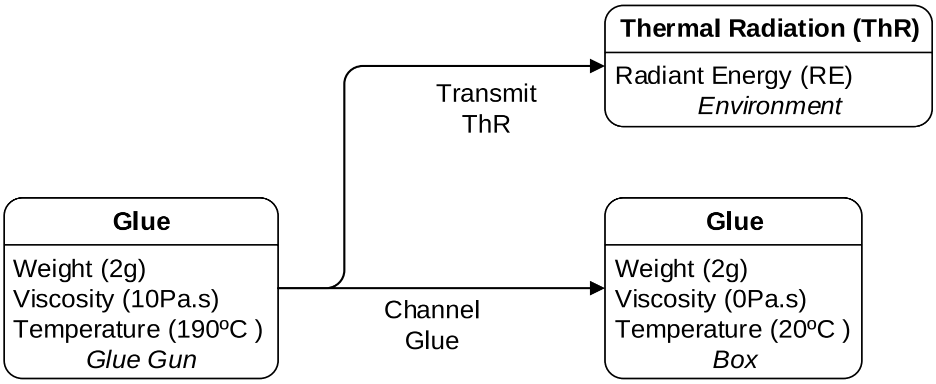


Figure 4.28: The flow of thermal radiation through the glue gun

Figure 4.29 shows an updated SSFD which includes the flow of thermal radiation branching out of the glue gun main flow.

As shown in Figure 4.29, the gun directs the flow of thermal radiation outside of the system boundary (i.e. transmit thermal radiation). The flow of the radiation braches out before the function text “channel glue”, since it reflects the flow of the radiation while the hot glue is in the glue gun. For the law of conservation of energy and material, the input and the outputs in respect of the function “channel glue” in Figure 4.29 should be equal as described by the equation “Glue(2g,10Pa.s,190ºC,Glue gun)=ThR (RE,Environment)+ Glue(2g,0Pa.s,20ºC,Box)”.

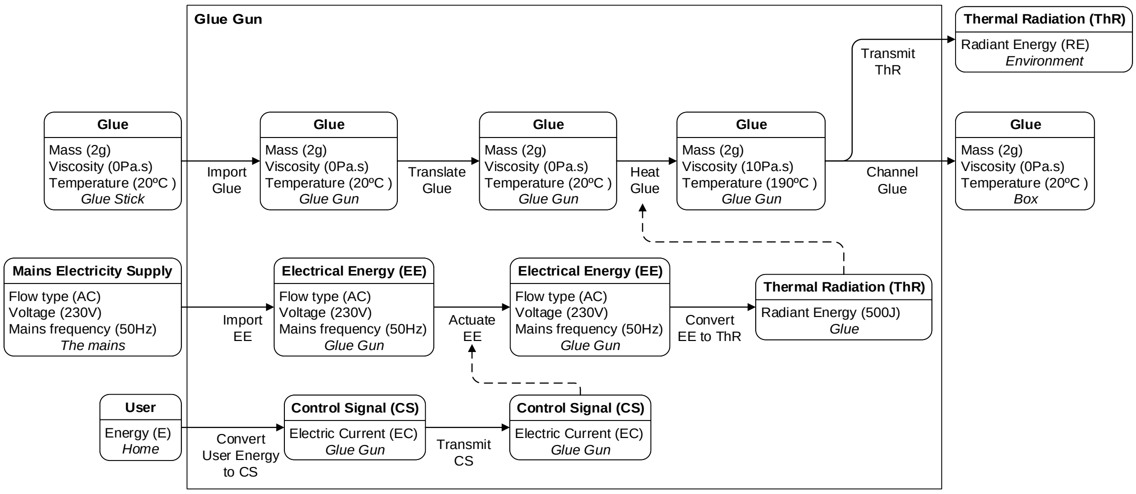


Figure 4.29: SSFD for the glue gun: the flow of thermal radiation branching out of the main flow

The main difference between the toaster and the glue gun case studies is that the glue gun SSFD is indirectly related to the user by being focused on the state of a box in respect of the user requirement “closing a box”, as shown in Figure 4.19. The connecting (i.e. energy) flow of the glue gun has the same characteristics of the toaster`s, however its branching flow is different. The gun addresses the flow of energy (i.e. thermal radiation) branching out of the main flow.

4.2.2 Radiant Heater

The flow of energy is represented as connecting flows in the toaster and the glue gun case studies. The latter also represents the flow of energy as a branching flow, as shown in Figure 4.29. This section focuses on the development of SSFD of a system whose main flow is energy.

A heating device warms the environment in which it operates. The environment may be the user of the device, an object, etc. Of the common means of heat transfer illustrated in Figure 4.5, a heating device usually works through convection or radiation. For example, a fan heater heats air at room temperature and directs the warm air towards the user and the surrounding space. The main flow of the device is the flow of air in this case. In terms of radiation, a radiant heater generates thermal radiation from a given energy source. Thermal radiation travels through air until it is absorbed by an object (e.g. user) and therefore the main flow of the device is energy.

In this case study, the design solution for warming the environment is assumed to be based on a household radiant heater with various heat settings. A radiant heater is generally used to warm the user; therefore, thermal radiation generated by the heater should interact with the user to warm the user, as illustrated in Figure 4.30.

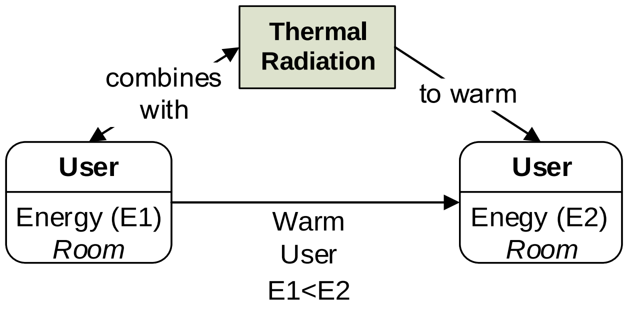


Figure 4.30: The triad of the input state “user”, the output state “user” and the output of the radiant heater main flow “thermal radiation” for the function “warm user”

The main flow of the heater should generate thermal radiation from a given energy source. Like the bread toaster and the glue gun case studies in the previous sections, mains electricity can be used as energy source, which is converted into thermal radiation. Figure 4.31 shows a high-level SSFD for the main flow in a radiant heater. The value of electric current is denoted by “EC”, e.g. EC1, EC2, etc.

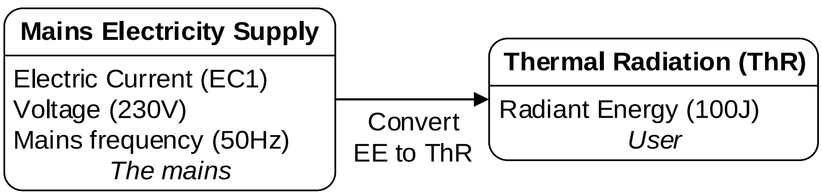


Figure 4.31: A high-level SSFD for the main flow in a radiant heater

Figure 4.31 shows the conversion of electrical energy in the mains into thermal radiation at the user. Considering the design requirement of the heater (i.e. heat settings), conversion and operation operations during the state transition in Figure 4.31 are represented in Figure 4.32.

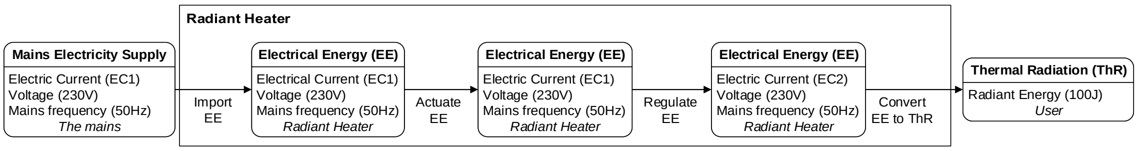


Figure 4.32: The main flow through the radiant heater

The diagram in Figure 4.32 begins with electrical energy import from the mains (i.e. import EE). Once the flow of electrical energy within the device is commenced (i.e. actuate EE), the level of radiant heat can be adjusted (i.e. regulate EE). This follows the conversion of electrical energy into thermal radiation (i.e. convert EE to ThR).

The functions “actuate EE” and “regulate EE” in Figure 4.32 perform in response to imported control signals. A household radiant heater can control the mode of the device (i.e. actuate EE) and the level of radiant heat (i.e. regulate EE) by providing manual control via the user or automated control based on process measurements, e.g. sensing room temperature. In terms of manual control, these controls signals are generated with respect to user inputs. The heater should convey relevant user requirements (i.e. the mode of the device and the level of radiant heat) by converting the user inputs to control signals and associating these control signals with the functions “actuate EE” and “regulate EE” in Figure 4.32. Figure 4.33 shows the process of converting a user input (e.g. energy) into a control signal in the heater for the functions “actuate EE” and “regulate EE”

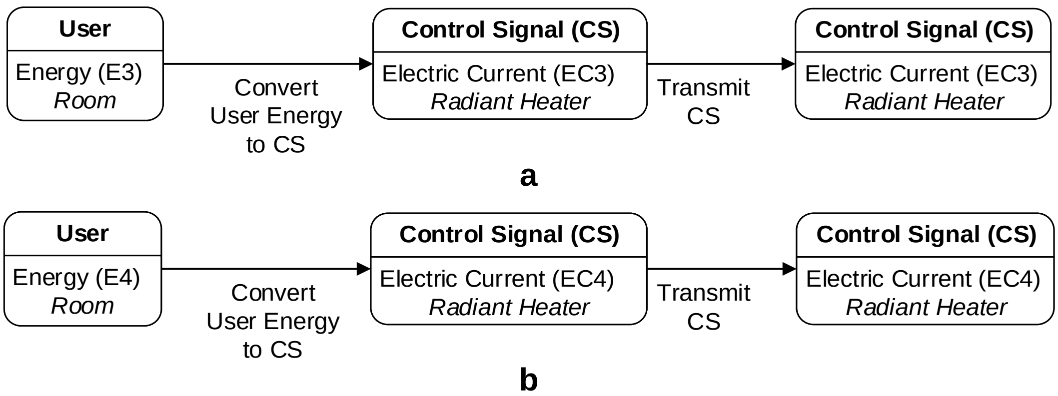


Figure 4.33: The flow of information through the heater for the functions “actuate EE” (a) and “regulate EE” (b)

As shown in Figure 4.33, attribute values for each information flow are different. The functions “actuate EE” and “regulate EE” on the main flow require the output states of the information flow (a) and (b) in Figure 4.33, respectively. These states combine with the inputs states of these functions, as shown in Figure 4.34.

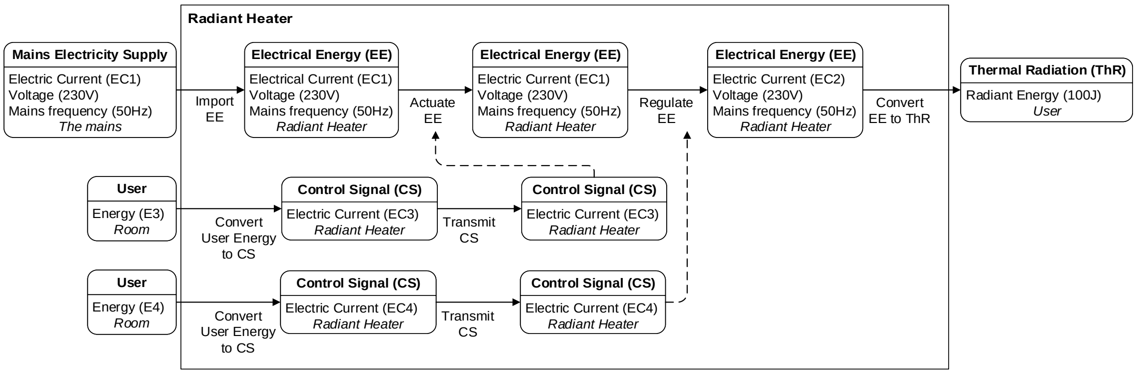


Figure 4.34: SSFD for the radiant heater: the flow of information

Considering the nature of the flow of thermal radiation, part of thermal radiation generated by the heater main flow is partially transmitted to the user. Part of the radiation go towards objects in the surrounding space. Figure 4.35 shows the branched flow of the radiation.

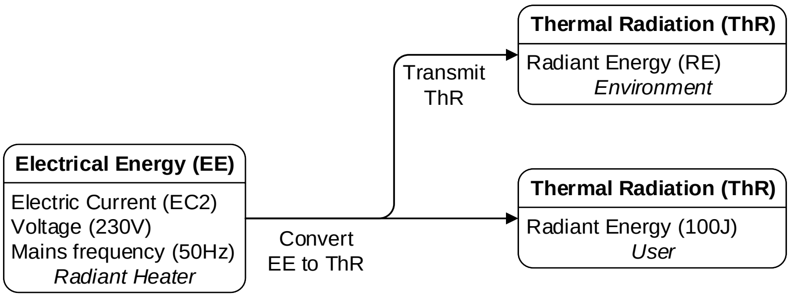


Figure 4.35: The flow of thermal radiation through the heater

The radiant energy value of the branched thermal radiation is denoted by “RE”. According to the law of conservation of energy, the input and the output in Figure 4.35 should be equal and this can be described by the equation “EE(EC2,230V,50Hz,Radiant Heater)=ThR (RE,Environment)+ThR (100J,User)”. Figure 4.36 shows an updated radiant heater SSFD which includes the flow of thermal radiation branching out of the heater main flow.

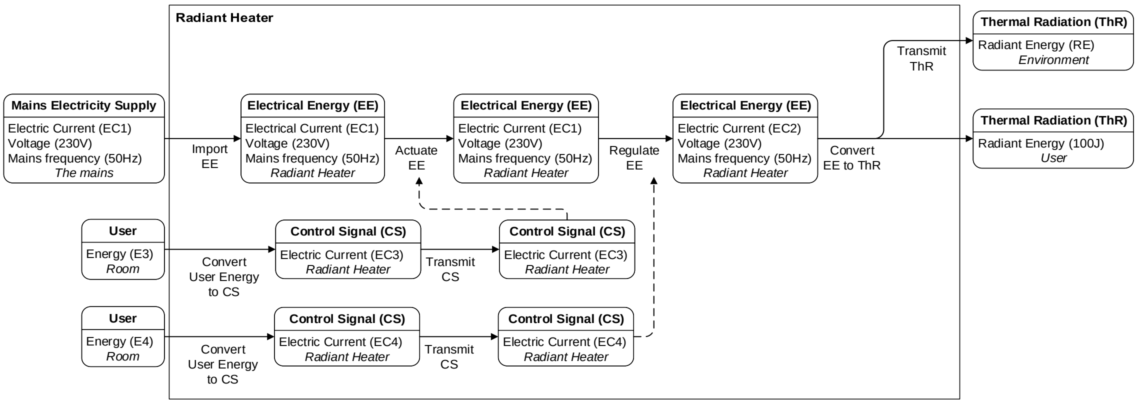


Figure 4.36: SSFD for the radiant heater: the flow of thermal radiation branching out of the main flow

Radiant heater SSFD in Figure 4.36 represents the flow of thermal radiation converted from electrical energy in the mains as the main flow, while the same flow is shown by the toaster SSFD in Figure 4.18 and the glue gun SSFD in Figure 4.29 as a connecting flow. The analysis of the flow of energy as a main flow in the design of the radiant heater required the determination of the branching flow out of the flow of energy, however the identification of the same flow in the toaster and the glue gun SSFDs requires the analysis of the flow of energy as the main flow.

4.2.3 Fuel Gauge

This section represents the development of SSFD of a system whose main flow is information. Functionality of a fuel gauge can be described as to measure and indicate the contents of any storage tank. Fuel contained in the fuel tank of a motor vehicle will be under consideration in this case study. The intended effect of the gauge on the driver is to make the contents of the fuel tank known to the driver. This requires the gauge to produce a signal to keep the driver informed, as shown in Figure 4.37.

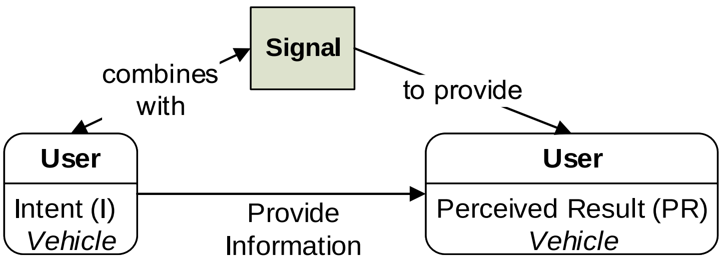


Figure 4.37: The triad of the input state “user”, the output state “user” and the output of the gauge main flow “signal” for the function “provide information”

The values of the user`s intent (i.e. a need for information regarding fuel level in fuel tank) and perceived result (i.e. fuel level in fuel tank is known) are denoted by “I” and “PR”, respectively. The design solution for this case study is assumed to be generating the signal in Figure 4.37 by processing an existing signal in respect of the contents of the fuel tank. Figure 4.38 shows a high-level SSFD for the main flow in a fuel gauge.

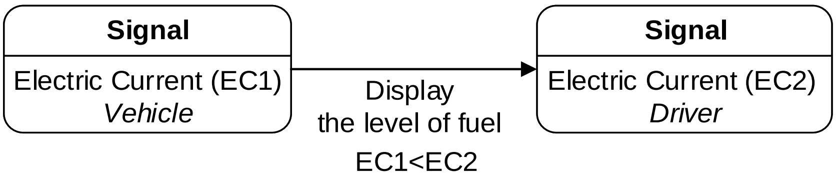


Figure 4.38: A high-level SSFD for the main flow in a fuel gauge

The value “EC1” of electric current of the input signal denotes the current flows through the vehicle as soon as the vehicle is turned on. In this case, the least current is flowing and the level of the fuel tank seems empty. The value “EC2” of electric current of the output signal shows the level of the fuel which is displayed to the driver. Once a signal regarding the level of fuel in the fuel tank is received (i.e. sense signal), it is indicated to the driver. Figure 4.39 shows the main flow through the fuel gauge.

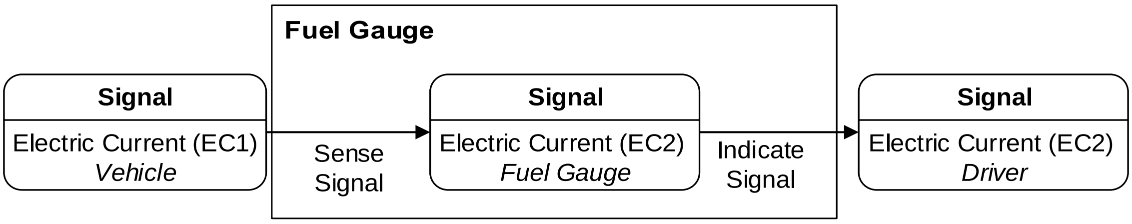


Figure 4.39: The main flow through the fuel gauge

The function “sense signal” in Figure 4.39 requires a signal regarding the level of fuel in the fuel tank. This signal combines with the input state of the function “sense signal” to alter the input signal in respect of the fuel content. The signal regarding the level of fuel can be generated by converting the fuel content in the fuel tank into a signal and linking this signal with the function “sense signal”. Figure 4.40 show the process of converting the fuel content in the fuel tank into a signal in the fuel gauge.

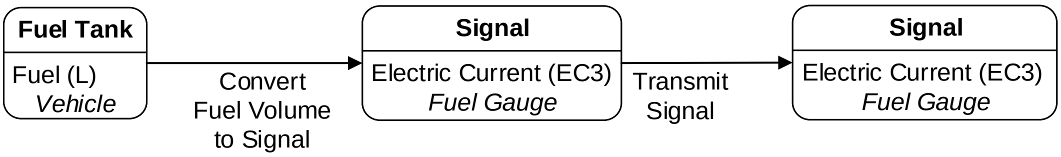


Figure 4.40: The flow of information through the gauge

The function "sense signal" in Figure 4.39 requires the output of the information flow in Figure 4.40. Figure 4.41 shows an updated fuel gauge SSFD aggregating the information flow in Figure 4.40 into the main flow in Figure 4.39. The value of fuel level in the tank and the value of electric current of its corresponding signal are represented by "L" and "EC3", respectively.

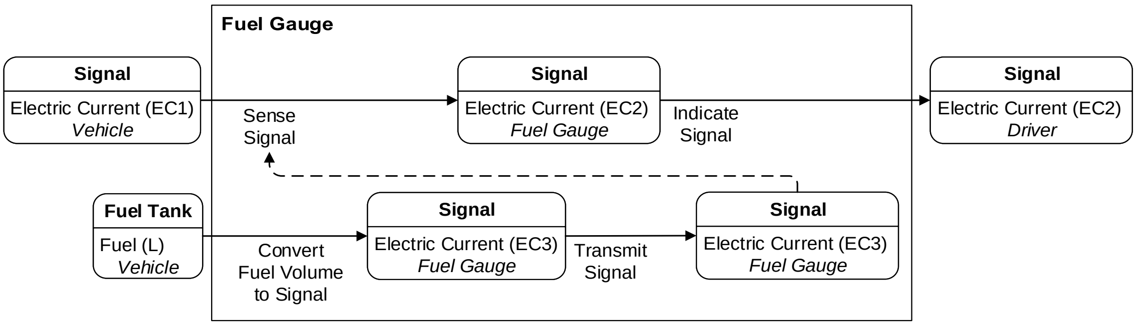


Figure 4.41: SSFD for the fuel gauge: the flow of information

A supply of external energy is required for the function "convert fuel volume to signal" whose output state is related to the fulfilment of the function "sense signal". Fuel gauge is a sub-system of a motor vehicle. Some design decisions during the development of the functional model of the gauge should be made in respect of the design of the vehicle. Therefore, it can be suggested that the vehicle provides low voltage electrical energy to the fuel gauge. Figure 4.42 shows the flow of electrical energy from the vehicle to the fuel tank.

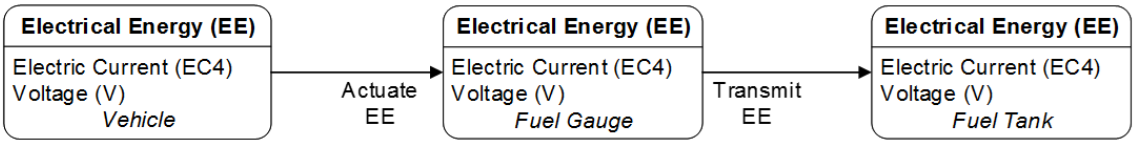


Figure 4.42: The flow of energy through the fuel gauge

Figure 4.42 shows that the flow starts by commencing the flow of electrical energy (i.e. actuate EE). Next, it is transmitted to the fuel tank in respect of the function "convert fuel volume to signal". Figure 4.43 shows an updated fuel gauge SSFD which includes the flow of energy through the fuel gauge.

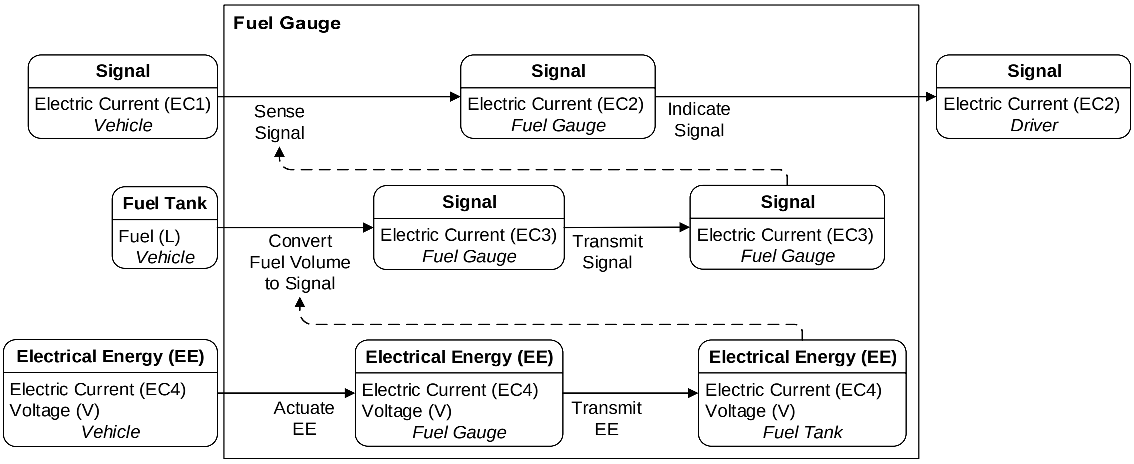


Figure 4.43: SSFD for the fuel gauge: the flow of energy

The term "signal" is represented as an object in Figure 4.43, as well as in the other case study SSFDs. For Pahl et al. (2007), the term "signal" is a more concrete expression of information e.g. control impulse, data and magnitude. While the signal is used as the general concept of information in many engineering systems, the existence of information can be articulated in different ways. As mentioned by Dori (2002), a human brain stores huge amounts of information that provides the basis for intelligence, e.g. decision making. Brain generates electric field as a result of an electrochemical process used by neurons for signalling. Hence, Figure 4.37 shows the user`s intent and perceived result as measurable attributes and their values are denoted by "I" and "PR", respectively.

4.3 Chapter summary

This chapter introduced the SSFD heuristics to conduct function modelling of a system using the SSFD function model. As represented in the beginning of Section 4.1, the heuristics are incrementally applied for function modelling of a system as follows:

- The first heuristic, the main flow heuristic, is used to determine the flow which is related to the intended effect of the system on the user.
- The second heuristic, the connecting flow heuristic, determines the flow(s) which branches into the main flow by specifying the flows of additional sources to the conversion operations on the main flow.

- The last heuristic, the branching flow heuristic, determines flow(s) that branches out of the main flow by focusing on each object attribute change on the main flow.

Section 4.1 illustrated these heuristics on a bread toaster, while the applicability of each heuristic to the flows of material, energy and information was tested on the desktop case studies “glue gun”, “radiant heater” and “fuel gauge” in Section 4.2.

5. Deployment of the SSFD to Develop Functional Model of Complex Multidisciplinary Systems

This chapter presents research to develop function modelling of systems with multiple operation modes by developing and extending the SSFD framework. The chapter starts by describing the research motivation and the methodology to develop an innovative method of function modelling on the basis of the SSFD framework. This follows the review of existing approaches in terms of modelling of multiple operation modes. The developed framework is illustrated on a front split view camera and further tested and validated on an electric vehicle powertrain and an active rear spoiler.

5.1 Motivation

Most of the approaches reviewed in Chapter 2 present function models of a system based on one mode of operation. Similarly, Chapter 4 showed the heuristics of the SSFD based on one mode of operation of the case studies, e.g. toasting a bread for the bread toaster. Observation of current systems engineering design practice through industrial collaborative research at the Bradford Engineering Quality Improvement Centre (BEQIC) has pointed that the prevalence of systems with multiple modes of operation (see also Liu et al., 2015), which each have different functional requirements. Therefore, the SSFD framework needs to be enhanced to support (1) the analysis and (2) the representation of the functional model of a system with multiple operation modes in a single diagram.

It is possible to hypothesise that the development of a SSFD for a system with multiple operation modes requires to develop a SSFD for each operation mode. Next, the developed SSFDs can be aggregated into a single SSFD by coupling functions and state flows of these SSFDs on the basis of shared states. By doing so, different operation modes of a system can be represented in a single diagram. This raises a question about how to ensure the coherency between different SSFDs. This chapter addresses this question by introducing "Enhanced Sequence Diagram (hereafter ESD)" methodology and "SSFD Fork Node" for the development of SSFD of a complex multidisciplinary system with multiple operation modes. The structure of the chapter is represented as follows:

- Section 5.2 summarizes how relevant approaches in Chapter 2 address the coupling of multiple functions and multiple flows. The second part of this section focuses on sequence diagrams.

- Section 5.3 introduces the ESD. Next, the process of developing a SSFD using the ESD for each use case (operation mode) of a system is represented as a three step process. A front view split camera (FVSC) of an automobile is used as an illustrative example for the representation of this process based on one use case.

- Section 5.4 focus on the development of systems with multiple operation modes. The ESD enhances the ability of the SSFD in the representation of flows through a system in a coherent way. This section introduces SSFD fork node to integrate SSFDs developed through ESDs for each operation mode of a system into a single diagram. The application of the enhanced sequence diagram and the SSFD fork node will be based on the analysis of an electric vehicle powertrain (EVP) and an active rear spoiler (ARS) which have been selected as a sample of complex multidisplinary systems with multiple operation modes in Section 3.3. There are three main engineering functional requirements of an electric vehicle powertrain (Campean et al., 2011), while an active rear spoiler has different operation modes depending on the position of the spoiler which changes in relation to the speed of the vehicle in order to manage the air flow around the vehicle towards useful functions, e.g. increased downforce lift at high speeds. As discussed in Section 3.3, the main difference of the ARS to the EVP is that the ARS addresses multiple operation modes by changing the global attribute “location” of the spoiler, while the EVP modifies “local attributes” of electrical energy with respect to the three functional requirements.

5.2 A critical review of literature on the analysis of systems with multiple operation modes

5.2.1 Established approaches in literature

As discussed in Section 2.3.2.1, the FFBD represents the top level system functions in a sequential relationship. Unlike the FFBD, Ullman (2010) suggested that a system may have different operating sequences which can be determined by rearranging its sub-functions. The same point is noted in the $C\&C^2$-A by using the term “sequence model” (Matthiesen and Ruckpaul, 2012). According to Albers et al. (2008), a sequence consists of at least two states and it determines the operational mode of a system. As discussed by Matthiesen and Ruckpaul (2012), a state can be part of different sequences and can address the fulfilment of several functions. Therefore, they suggest using sequence model for each dynamic system. Umeda et al. (1996) use the term “aspect” (see Section 2.3.5.2) to represent different behaviours of the same entity. Aspects are related with each other through states. Statecharts by Harel (1987) represents the functionality of a system in the same diagram, independent of each other. While Ulrich and Eppinger (2003) suggest determining system functions in terms of sequence of user actions in the analysis of systems involving numerous user interactions, UML/SysML use case diagram (see 2.2.6.5.1) is viewed as a way of capturing system requirements in terms of how the system is used by its users. The use cases can be related with each other in different ways (e.g. inclusion), but they are not represented in a particular order.

In the case of coupling multiple flows, the FFBD maps the flows of functions through a system and they are related to each other by arrows. Specific cases are represented by gates; “AND”, “OR”, “Go” or “No-Go”. Some diagrams include “exclusive OR (XOR)”, “iteration (IT)”, “repetition (RP)” and “loop (LP)” gates (NASA, 2007). The fractal character of the $C\&C^2$-A enables the approach to couple different operating sequences through its main elements. The approach addresses discrete states of a system by introducing the notion of “switch” with the states “on” and “off” (Matthiesen and Ruckpaul, 2012). As discussed in Section 2.3.6.5.1, a use case in SysML is often detailed through activity, state machine and sequence diagrams. The sequence of actions in activity diagrams can be specified using the nodes “decision”, “merge”, “fork” and “join”, while the nodes “choice”, “junction”, “fork” and “join” are used in the case of coupling states of state machine diagrams. The nodes “decision” and “choice” has the same notation. They refer to different operation modes of a system by having one incoming flow and several outgoing flows, which each have a condition. Sequence diagrams can be related to each other through interaction operators, i.e. alt, opt, break, loop, seq, strict, par operator, critical, neg, assert, consider and ignore (see Friedenthal et al., 2008).

5.2.2 Sequence diagrams

For Latronico and Koopman (2001), sequence diagrams can provide a basis for developing statecharts, while SysCARS (System Core Analyses for Robustness and Safety) methodology points out that the link between a use case diagram and a state machine diagram can be established through a sequence diagram (see Piques, 2014).

Sequence diagrams are based on the Message Sequence Charts (MSC) of the Specification and Description Language (SDL) (Weilkiens, 2006). Of UML interaction diagrams (i.e. sequence diagram, communication diagram, timing diagram and interaction overview diagram), SysML uses the sequence diagram. In addition to UML and SysML, sequence diagrams are used in other languages too. For example, Hoffman (2011) introduces the basic concepts of Harmony for Systems Engineering on the basis of SysML diagrams including sequence diagrams, while UML diagrams (including sequence diagrams) are used in object-oriented analysis and design (Booch et al., 2007).

Several researchers extended sequence diagrams to support function modelling of a system. Xie et al. (2009) reviewed the variants of the UML sequence diagram notation. One of these variants shows an object on its lifeline in terms of two discrete states “locked” and “unlocked”. Zingel et al. (2012) mentioned that sequence diagrams describe specific usage sequences of a system in terms of concrete events. They extended sequence diagrams to define functions for specific operation of a system. Figure 5.1 shows a simplified sequence diagram for a test case of a hybrid powertrain.

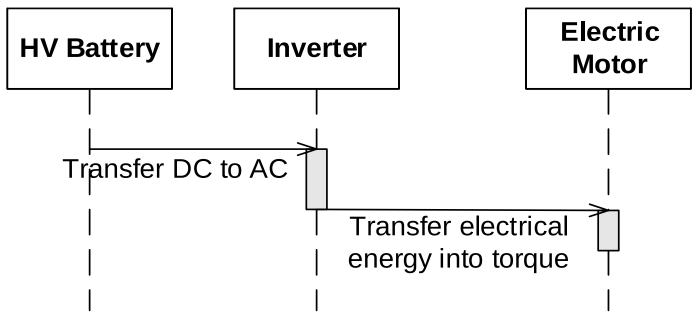


Figure 5.1: Sequence diagram for a test case (adapted from Zingel et al., 2012)

On the above figure, the performed functions are articulated as “transfer DC to AC” and “transfer electrical energy into torque”. HV battery, inverter and electric motor are the affected components. As shown in Figure 5.1, the diagram represents a function (i.e. triggered event) by an arrow. Similarly, Piques (2014) complemented the sequence diagrams by functions to be implemented by the system. Figure 5.2 shows sequence diagram for a hybrid vehicle at system level.

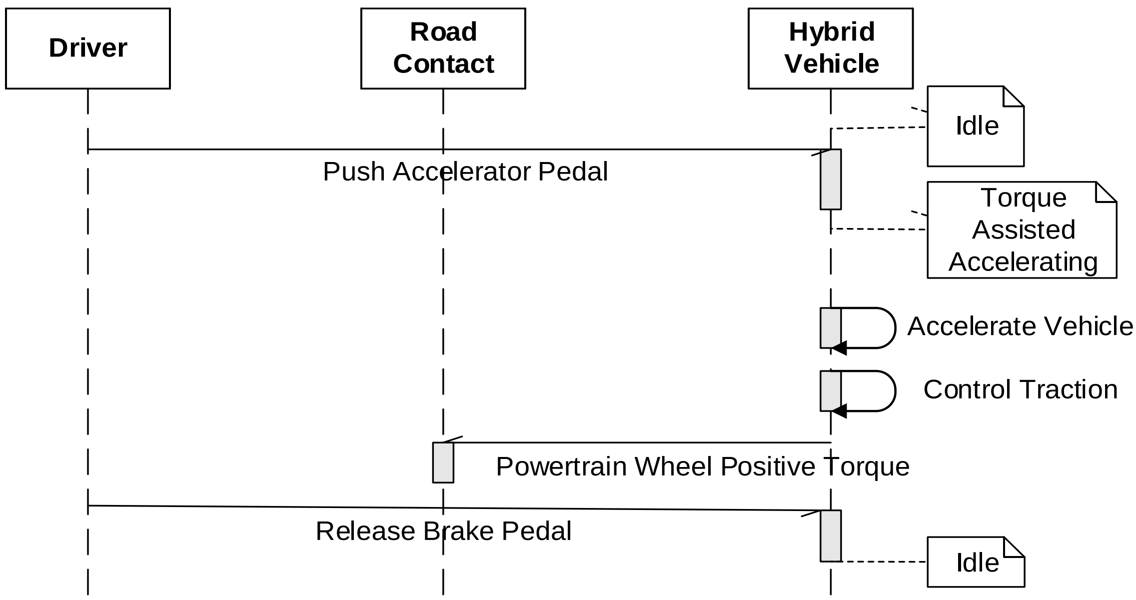


Figure 5.2 Sequence diagram for a hybrid vehicle (adapted from Piques, 2014, p.8)

As shown in Figure 5.2, the functions are shown in terms of SysML operations (i.e. events) attached to the lifelines of the blocks “driver”, “road contact” and “hybrid vehicle”. The starting point and ending point of each diagram will match to states of the system state machine diagram, while transition conditions between states of the system state machine diagram are determined through the interactions between the blocks on the sequence diagram (Piques, 2014).

The fact of including functions (Zingel et al., 2012; Piques, 2014) and object states (Xie et al., 2009) in sequence diagrams support function modelling; however, the question of how to apply sequence diagram in a system including multiple flows has not been addressed. This requires to adapt the sequence diagram to the SSFD concept, as represented in the following section.

5.3 Enhanced sequence diagram as a basis for the development of a SSFD

5.3.1 Approach

As discussed in Section 2.3.6.5.1, a sequence diagram is developed based on a use case diagram in SysML. This section adapts the methodology of Piques (2014) and shows how to develop a SSFD from a sequence diagram in respect of a specific system function. While the concepts of SysML use case diagram is used in the representation of the functionality of a system, SysML sequence diagram is adapted to the SSFD concept, called enhanced sequence diagram (ESD). Figure 5.3 represents the process of developing a SSFD using an ESD as a three step process.

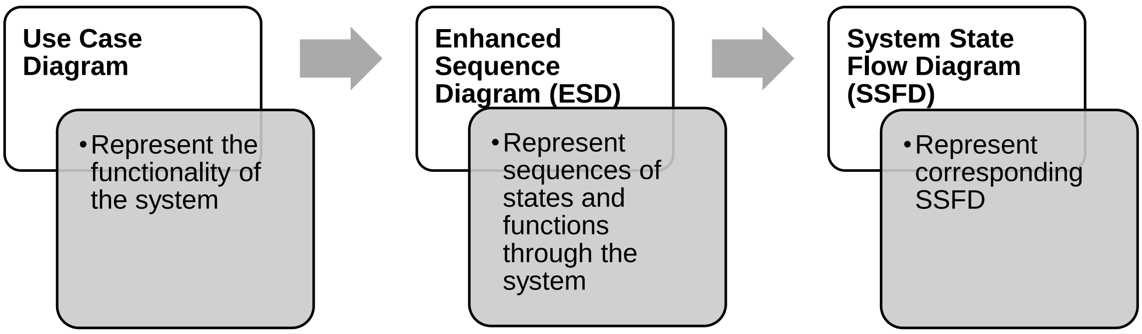


Figure 5.3 Overview of the process of developing a SSFD using an ESD

This section seeks to explain the development of the process in Figure 5.3 on a front view split camera (FVSC) of an automobile.

5.3.2 Front Split View Camera

5.3.2.1 Case study background

The Front Split View Camera (FSVC) case study is based on research work that has been conducted within the BEQIC in conjunction with a major automotive company. The FSVC is an advanced driver assistance system whose main functionality is to enable drivers to spot approaching objects (e.g. cyclist), that is, it addresses the main flow of information from the environment to the driver.

5.3.2.2 Use Case Diagram

A use case diagram represents the functionality of a system in terms of use cases that are required by actors, as discussed in Section 2.3.6.5.1. Figure 5.4 illustrates the main functionality of the FVSC as a use case with its actors.

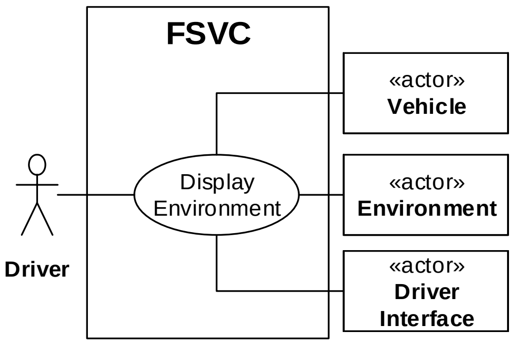


Figure 5.4: The main functionality of the FVSC as a use case diagram

As shown in Figure 5.4, the fulfilment of the use case “display environment” requires the actors “Vehicle”, “Environment”, “Driver Interface” and “Driver”. These actors interact with the FVSC (called the system under consideration in SysML) and may interact with each other in respect of this use case. Interactions between a system and its actors for a particular use case are identified through a sequence diagram.

5.3.2.3 Enhanced Sequence Diagram

This section extends SysML sequence diagram with different notions with the aim of supporting the development of a SSFD. This diagram is therefore termed Enhanced Sequence Diagram (ESD). Figure 5.5 shows the proposed schema of enhanced sequence diagram.

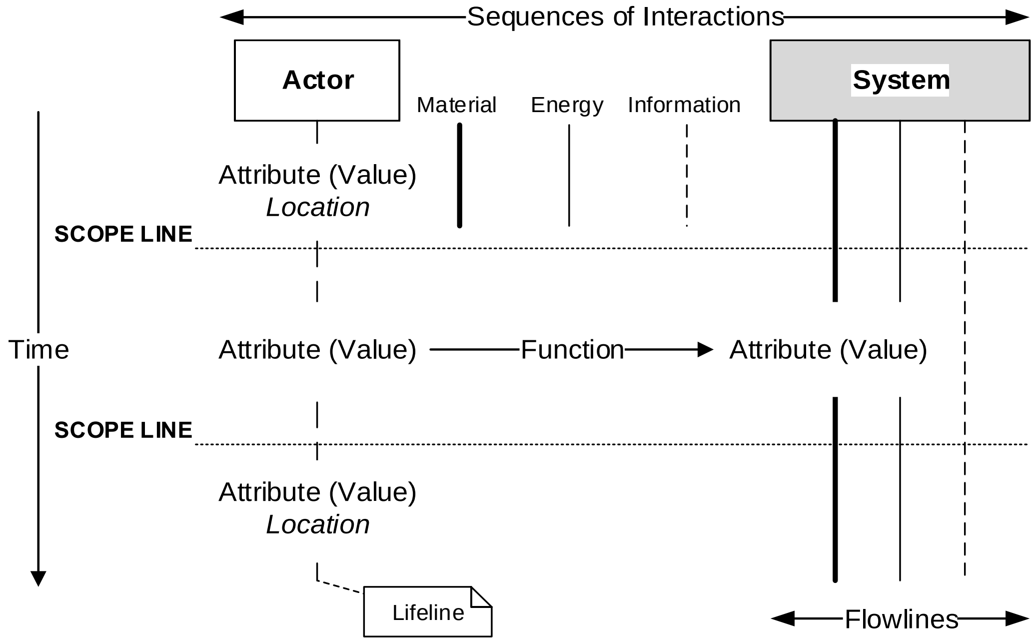


Figure 5.5: The proposed schema of Enhanced Sequence Diagram

The key features of the diagram can be described as follows:

- UML/SysML sequence diagrams (as well as Xie et al. (2009), Zingel et al. (2012); Piques (2014)) use the term “lifeline” (see Section 2.3.6.5.2) to represent how long the actor exists during the interactions. As mentioned in Section 3.2, a technical system may consist of the flows of material, energy or information. The term “flowline” is introduced in Figure 5.5 instead of the term “lifeline” to represent the flows of material, energy and information through the system. Like UML/SysML sequence diagrams, these flows are represented by vertical lines with respect to time, however they are differentiated by denoting them with different line types, as shown in Figure 5.5. The term “lifeline” of UML/SysML sequence diagrams is used to represent the existence of the actor with measurable attributes. The actor interacts with relevant flowline of the system, as represented in Figure 5.5. A system may include multiple flowlines of material, energy or information. Grey box in Figure 5.5 differentiates the system from the actor.

- SysML sequence diagrams focus on interactions between the elements of a system in terms of a sequence of message exchanges, while the diagrams of Zingel et al. (2012) and Piques (2014) describe functions in respect of the lifelines of the blocks. The diagram in Figure 5.5 places emphasis on the state of the actor and the system by representing them with measurable attributes. Coherent with the SSFD function model, attributes of the actors and the system are represented on their lines with related to time. A function is denoted by a filled arrow and is referred to an interaction in respect of relevant attribute of the actor/system, as shown in Figure 5.5. The function text is shown on the arrow for the purpose of practicality of the representation.

- Figure 5.5 introduces the notion of “scope lines” to define the boundaries of the diagram and they are shown by horizontal dashed lines. Above of the upper scope line and below of the lower scope line show the initial and the final attributes of the actors/the system, respectively.

The first step in the development of an enhanced sequence diagram is to identify what actors are related to a particular use case. As noted in the previous section, the use case “display environment” of the FSVC is related to the actors “Vehicle”, “Environment”, “Driver Interface” and “Driver”. The enhanced sequence diagram in Figure 5.6 for the use case “display environment” represents interactions between the FSVC and these actors.

Unlike the development of a SSFD, the development of an enhanced sequence diagram starts by representing design decisions regarding the system in terms of actors. This requires to specify the initial and the final attributes of relevant actors, as shown in Figure 5.6.

- The actor “vehicle” provides electrical energy (EE) whose attribute value is denoted by “EE1”.

- The actor “driver” has two lifelines:

    - The lifeline on the left shows the intended effect of the FSVC on the driver in terms of the initial state “intent” (i.e. a request to view the environment from vehicle) and the final state “result” (i.e. the awareness of the objects in the environment).

    - The initial attribute value (DI1) on the other lifeline of the driver addresses the need for actuating the FSVC.

- The working principle of the FSVC is associated with enabling the driver to spot objects in the environment. This is provided by the actor “driver interface” which can be a screen or a speaker depending on the way of making the objects in the environment known to the driver. Figure 5.6 represents the initial and the final attributes of the driver interface as image status (IS) and current (C) whose values are related to the state of the driver.

- The actor “environment” is denoted by the term “object” in Figure 5.6. An object can be anything around the vehicle, and specified by its size (S) and location on the road. Locations of the actors in Figure 5.6 are written in italic.

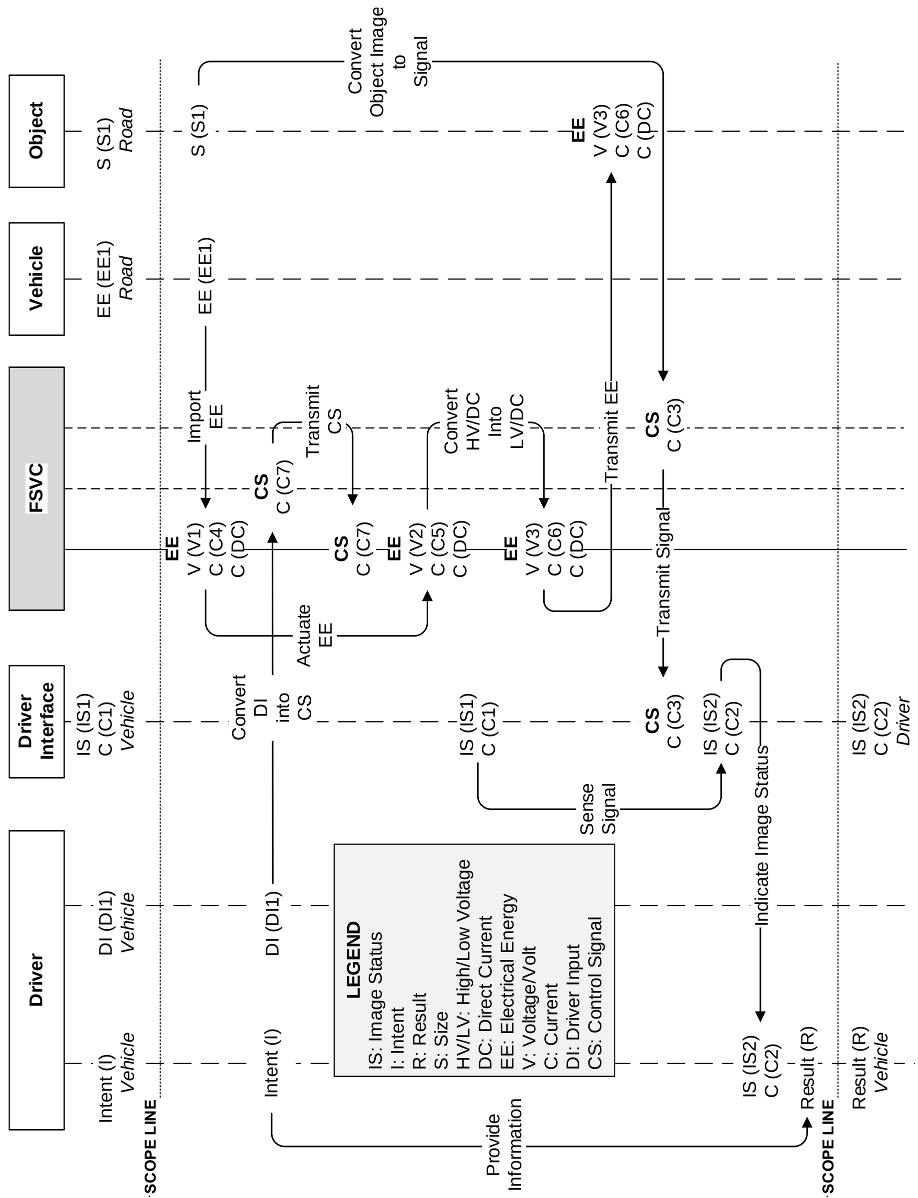


Figure 5.6: An ESD for the use case “display environment”

Once the initial and the final attributes of the actors are specified, interactions between the actors and the FSVC can be mapped in vertical direction with related to time as follows:

- Figure 5.6 shows that the first interaction starts by importing electrical energy (EE1) from the vehicle to the FSVC (i.e. import EE).
- The energy "EE" in the FSVC is actuated (i.e. actuate EE) once the driver input (DI1) is converted into a control signal (C7) (i.e. convert DI into CS) which is transmitted to the flow of EE in the FSVC (i.e. transmit CS).
- High voltage electrical energy (V2) is converted into low voltage (V3) electrical energy (i.e. convert HV/DC into LV/DC).
- Low voltage electrical energy (V3) is used (i.e. transmit EE) to convert object image (S1) in the environment into an electronic signal (C3) (i.e. convert object image to signal). The electronic signal is transmitted to the driver interface (i.e. transmit signal).
- The electronic signal "C3" changes the status of the driver interface (i.e. sense signal) by altering its attributes "image status (IS1)" and "current (C1)".
- The driver interface makes the presence of the object known to the driver (i.e. indicate image status) and this changes the state of the driver (i.e. provide information), as shown in Figure 5.6.

5.3.2.4 System State Flow Diagram based on ESD

The sequence diagram in Figure 5.6 represents sequences of functions and states through the FSVC. This section aims to represent corresponding SSFD based on the diagram in Figure 5.6.

The development of a SSFD starts by identifying the main flow through the system and continues determining the connecting and branching flows on the basis of the analysis of the main flow. An enhanced sequence diagram is developed in a reverse order. The enhanced sequence diagram maps interactions between the actors and the system top-down in related to time. This requires to develop the connecting flows at first and to represent the main flow in related to the intended effect of the system latest.

Flowlines of the FSVC in Figure 5.6 shows that the flows of energy and information go through the FSVC. Different flowlines for the flow of information show that they are related to different flows in the SSFD. Actors on a sequence diagram are denoted as objects on a SSFD. Similarly, attributes of the actors on the sequence diagram are shown as attributes of the objects on the SSFD. Each interaction between the actors and the system on the enhanced sequence diagram are represented as a function with a state transition on the SSFD. Figure 5.7 shows the conversion of a chain of interactions on the sequence diagram in Figure 5.6 into a state flow diagram.

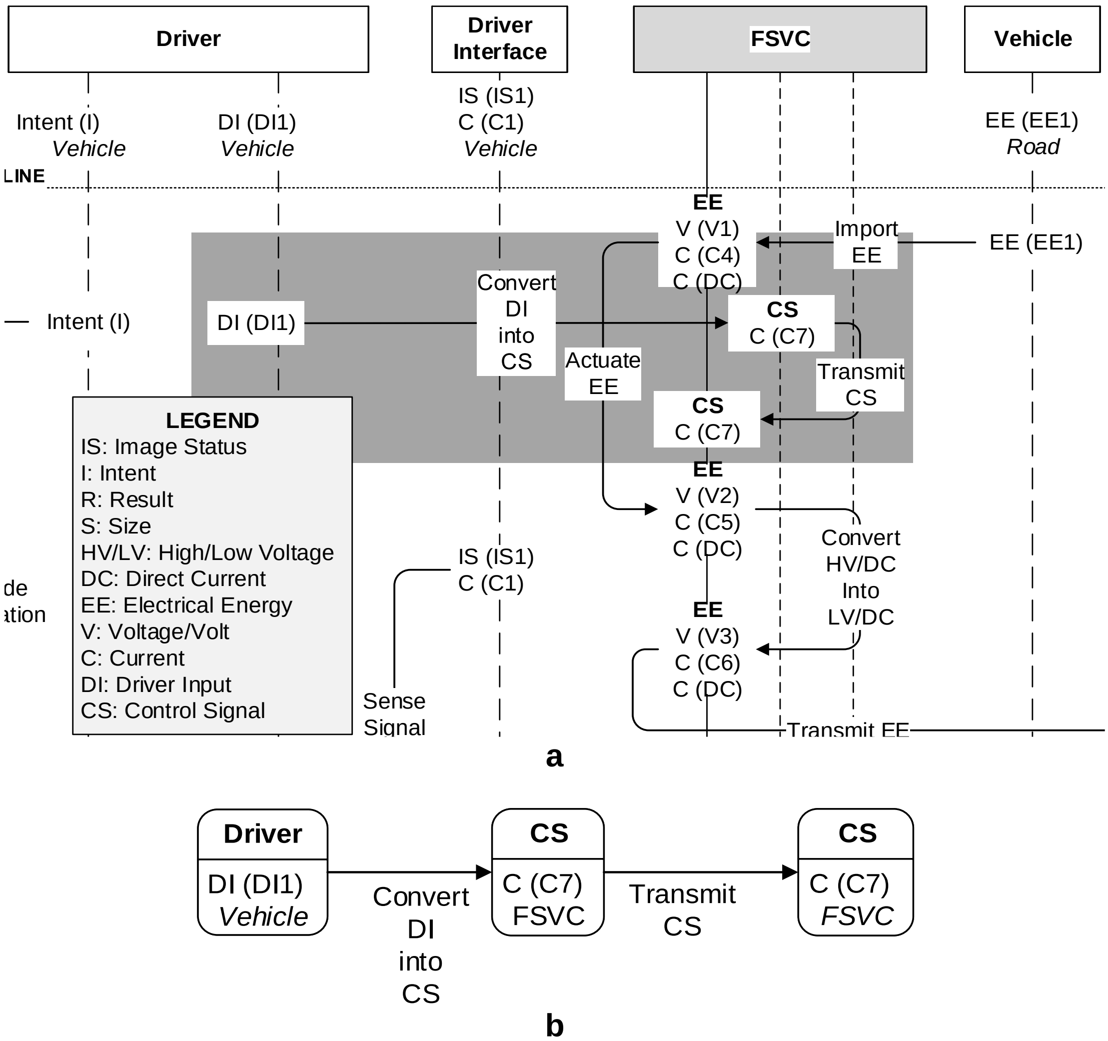


Figure 5.7: (a) an excerpt from the enhanced sequence diagram in Figure 5.6, (b) a state flow diagram based on the excerpt

Figure 5.7-b shows corresponding SSFD for the chain of interactions shown in grey rectangle in Figure 5.7-a. As shown in Figure 5.7-a, the global attribute “location” of the actors are not included within the scope lines for the clarity of the diagram. The output location of the control signal (CS) in Figure 5.7-b is shown as “FSVC” due to the fact that it is connected with the flow of electrical energy of the FSVC for the achievement of the function “actuate EE”. Figure 5.8 shows the complete state flow diagrams based on the sequence diagram in Figure 5.6.

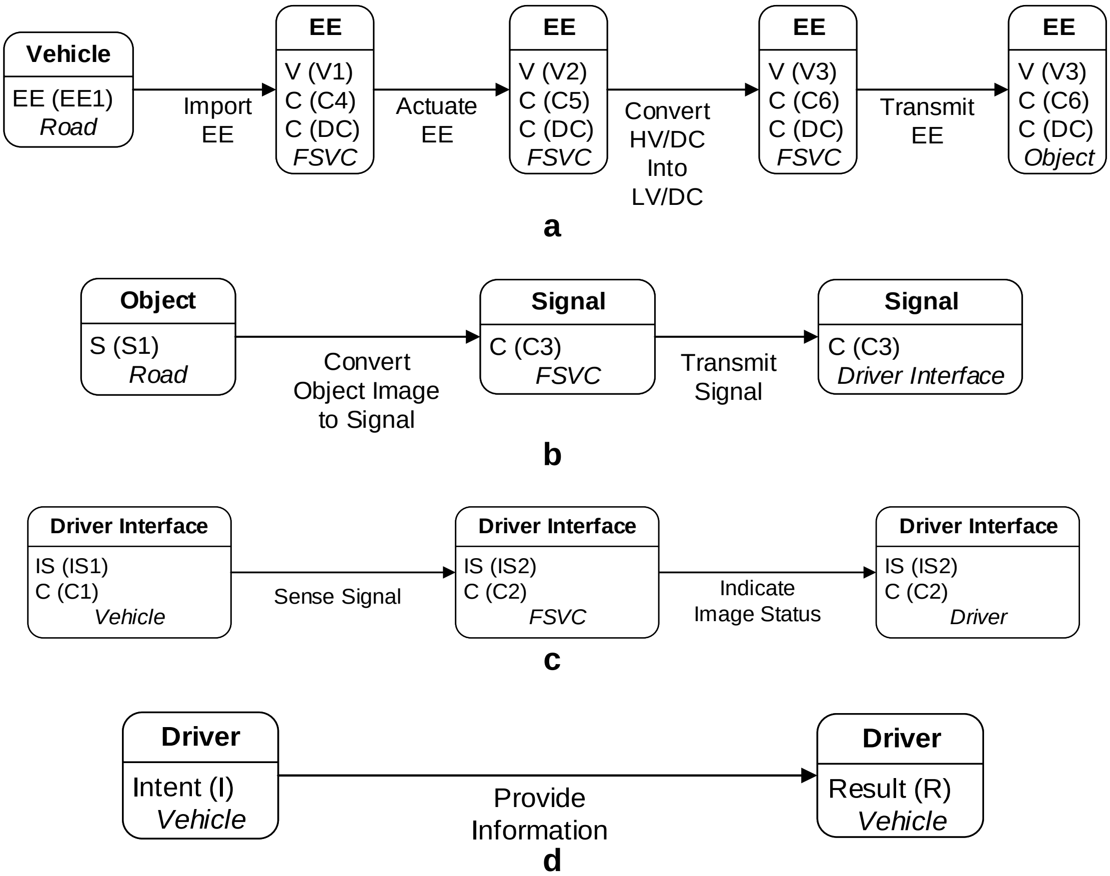


Figure 5.8: State flow diagrams extracted from the ESD in Figure 5.6

Coherent with the SSFD function model, the output of each flow in Figure 5.8 is the design solution for relevant conversion function of other flow. The links between state flow diagrams in Figure 5.8 can be described based on the enhanced sequence diagram in Figure 5.6 as follows:

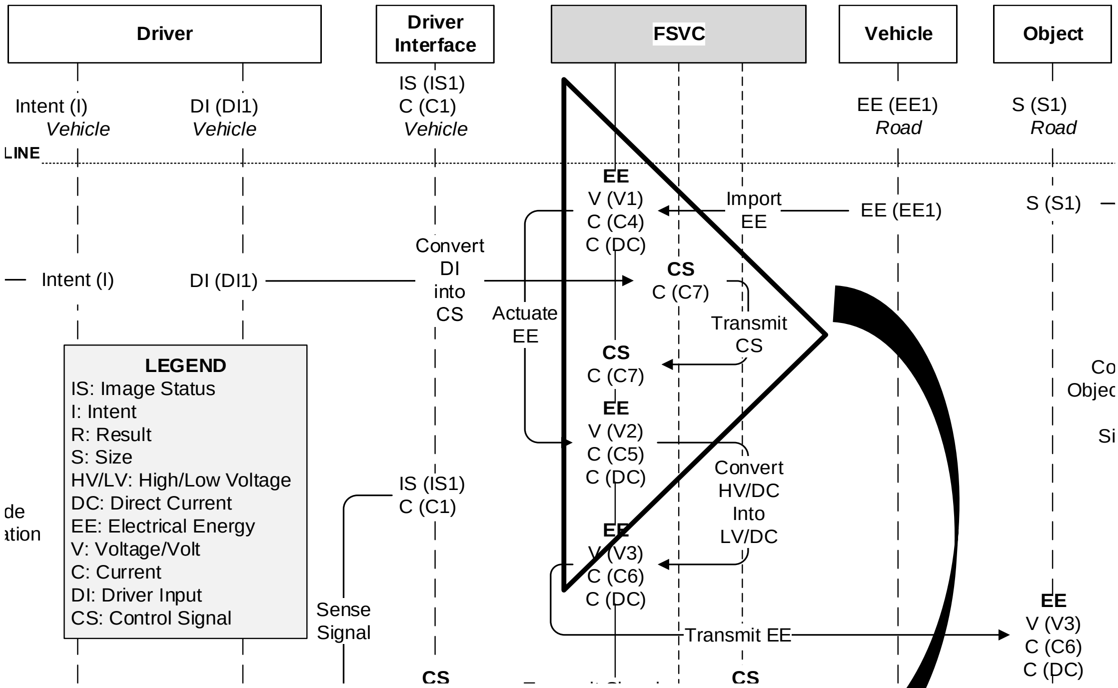
- The output state of Figure 5.7-b combines with the input state of the function “actuate EE” in Figure 5.8-a.

- The output state of Figure 5.8-a combines with the input state of the function “convert object image to signal” in Figure 5.8-b.

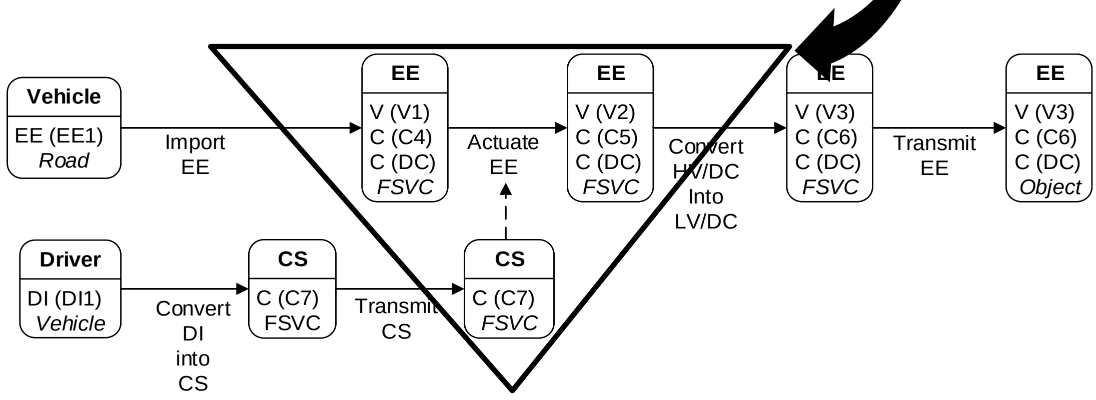
- The output state of Figure 5.8-b combines with the input state of the function “sense signal” in Figure 5.8-c.

- The output state of Figure 5.8-c combines with the input state of the function “provide information in Figure 5.8-d.

Figure 5.9 represents the first bullet point above, that is, the combination of state flow diagrams in Figure 5.7-b and Figure 5.8-a in respect of interactions on the sequence diagram in Figure 5.6.



a



b

Figure 5.9: (a) an excerpt from the enhanced sequence diagram in Figure 5.6, (b) a combined state flow diagram based on the excerpt

Figure 5.9-b shows the flow of electrical energy (EE1) from the vehicle and the conversion of driver input (DI1) into a control signal (C7) in relation to the fulfilment of the function “actuate electrical energy (EE)”. Location attribute “FSVC” in italic shows the flows of states within the FSVC. The final location “object” of the electrical energy reflects the fact that the energy is connected with the input state of the function “convert object image to signal” in Figure 5.8-b. Location attributes “FSVC” and “object” can be replaced with relevant design elements as soon as the system is designed. As discussed in Section 3.1.2.2., for the SSFD function model, a function is defined in terms the triad of an input state, an output state and a design solution. According to this model, the control signal “CS” combines with the input state of the function “actuate EE” to fulfil the function “actuate EE”. The “triad” is shown in a triangle in Figure 5.9-a and Figure 5.9-b.

By following the same methodology, all state flow diagrams of the FSVC can be aggregated into a single diagram, as shown in Figure 5.10.

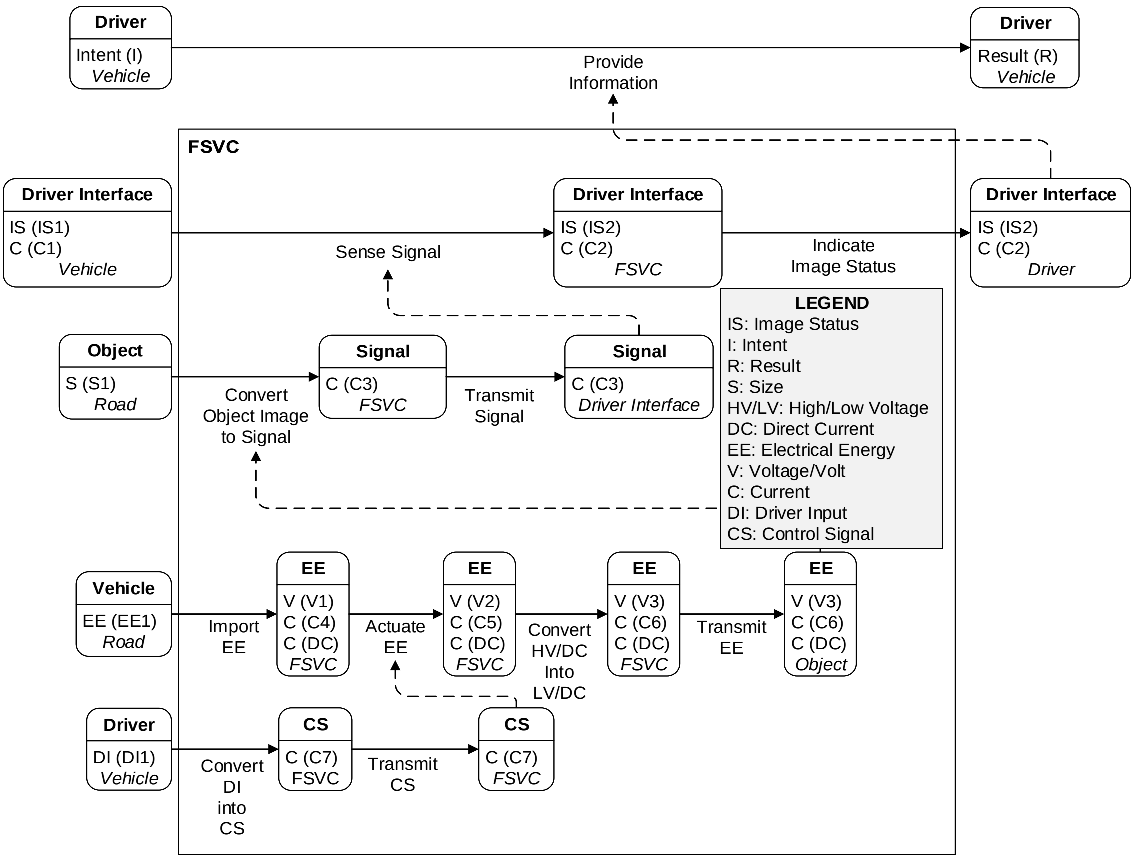


Figure 5.10: SSFD for the FSVC

The intended effect of the FSVC on the driver through the function “provide information” is related to the environment-centric view of the FSVC SSFD. The device-centric view of the FSVC SSFD represents the functionality of the FSVC in related to the environment-centric view. The device-centric view is differentiated from the environment-centric view by showing the device-centric view in a box which defines the “system boundary”, as shown in Figure 5.10.

5.3.3 SSFD and ESD - summary of key points

This section presented a three step process for the development of a SSFD by adapting the methodology of Piques (2014). The process steps are based on the development of three diagrams in the following order:

1) **Use Case Diagram:** The functionality of a system was represented using the concepts of SysML use case diagram in this section. For the purpose of illustrating the methodology, this section focused on one operation mode of the FSVC.

2) **Enhanced Sequence Diagram:** This section proposed the enhanced sequence diagram in related to the SSFD framework. The main functionality of the enhanced sequence diagram is to represent sequences of states and functions through a system with related to time.

3) **System State Flow Diagram:** The development of the SSFD through the SSFD heuristics in Chapter 4 focused on the achievement of relevant functional requirement of a system, i.e. firstly, the main flow is developed and so on. The development of a SSFD based on an enhanced sequence diagram enhances the ability of the SSFD to represent the flows through a system in a coherent way by taking into account sequences of states and functions through the system, i.e. the connecting flows are developed first and the main flow is developed latest. The basic elements of the SSFD and the ESD are alike, e.g. objects and actors are represented with measurable attributes. Therefore, representing a corresponding SSFD based on an ESD is straightforward.

The coherency between state transitions of different SSFDs of the same system can be provided by deploying this three step process for the development of the

SSFDs. The next section seeks to explain the deployment of this process to develop SSFD for systems with multiple operation modes.

5.4 Validation: The use of the ESD in the development of the SSFD for systems with multiple operation modes

5.4.1 Approach

The previous section focused on the development of the SSFD based on one operation mode by following the three step process summarized in Section 5.3.3. The objective of this section is to validate this process in the development of SSFD for systems with multiple modes of operation, which each have different functional requirements. The validation will be carried out based on real world case studies collaborated with industrial partners. As mentioned in Section 5.1, the process steps will be implemented on an electric vehicle powertrain (EVP) and an active rear spoiler (ARS).

5.4.2 Electric Vehicle Powertrain

5.4.2.1 Case study background

This case study based on collaborative research work completed within the BEQIC with an automotive engineering organization. The case study has been outlined in Campean et al. (2011). The original case study focused on interface identification and characterization of an electric vehicle powertrain (EVP) for a full electric light commercial vehicle at the system level using some known Failure Mode Avoidance (FMA) process tools such as State Flow Diagram, Boundary Diagram and Interface Analysis. For the purpose of this section, the case study will be reconsidered with a focus on function analysis of the electric powertrain using the proposed approach in Section 5.3.1.

The EVP controls the main flow of electrical energy through the vehicle to address three main engineering functional requirements which are described by Campean et al. (2011) as follows:

1) to charge and store electrical energy,

2) to provide controlled torque at the rear axle,

3) to provide power for low voltage vehicle consumers.

The development of the SSFD for the EVP will be based on these functional requirements. An EVP may operate under different circumstances, namely the sequence of its functional requirements may be varied. This section will analyse the functional requirements of the EVP by following the presented order above, i.e. the analysis will start with the functional requirement “charge and store electrical energy”.

5.4.2.2 Use Case Diagram for EVP

The first step in the development of a SSFD for a system with multiple functional requirements is to represent the functionality of the system as a use case diagram. Figure 5.11 shows some of the high level functionality of the EVP in terms of a use case diagram.

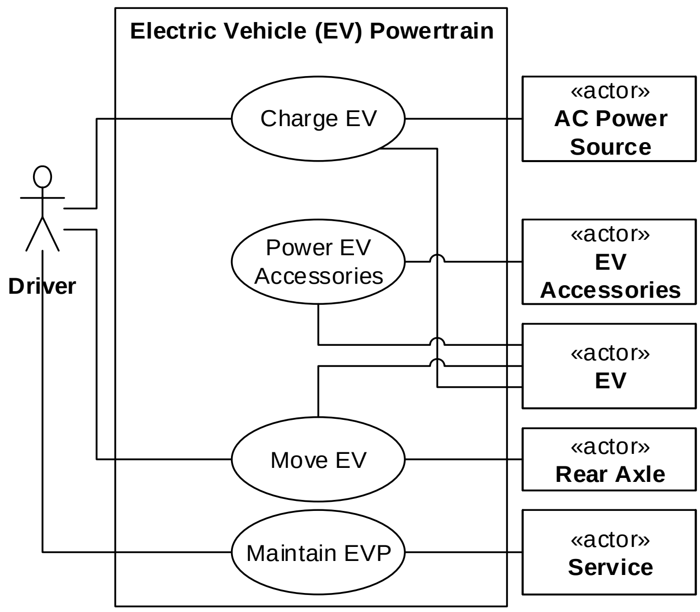


Figure 5.11: A set of use cases for the EVP

Use cases related to the EVP functional requirements described in the previous section are the focus of this section: charge EV (electric vehicle), move EV and power EV accessories.

5.4.2.3 Enhanced Sequence Diagram for EVP

This step aims to describe interactions between the EVP and its actors for the use cases “charge EV”, “move EV” and “power EV accessories” using an enhanced sequence diagram.

Figure 5.11 shows that the use case “charge EV” is linked to the actors “AC Power Source”, “Driver” and “Electric Vehicle”. Figure 5.12 details these links on an enhanced sequence diagram.

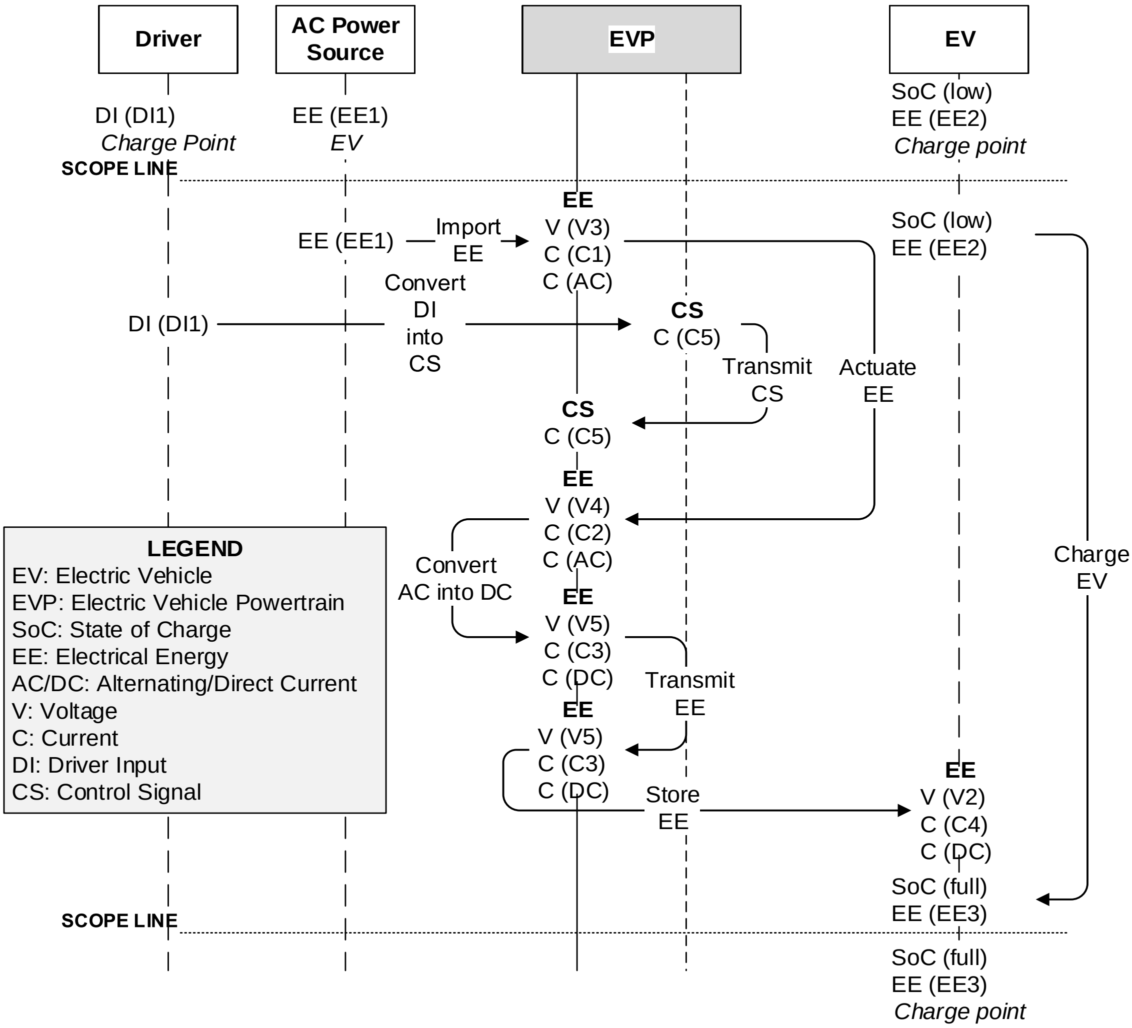


Figure 5.12: ESD for the use case “charge EV”

The sequences of interactions in Figure 5.12 can be summarized as follows:

- Driver input (DI1) is converted into a control signal (C5) (i.e. convert DI into CS) and the signal is transmitted to the EVP (i.e. transmit CS).
- As soon as the signal is transmitted to the EVP, the imported electrical energy (EE1) (i.e. import EE) is actuated (i.e. actuate EE) and then converted into Direct Current (DC) (i.e. convert AC into DC).
- The converted energy flows through the EVP (i.e. transmit EE) and stored in the EV (i.e. store EE) which means the EV is on charge (i.e. charge EV).

The use case “move EV” is related to the actors “Rear Axle”, “Driver” and “Electric Vehicle” in Figure 5.11. Figure 5.13 shows an enhanced sequence diagram for the use case “move EV”.

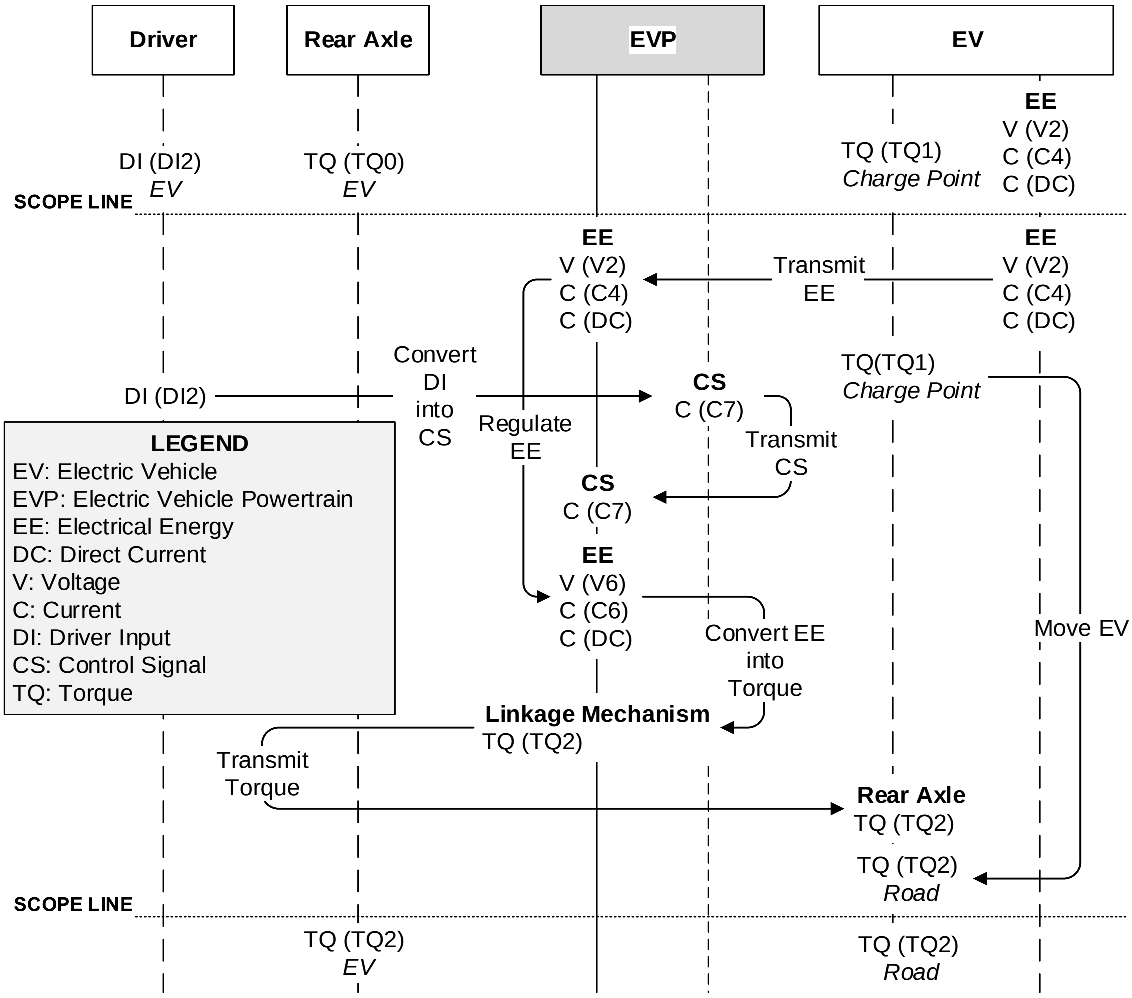


Figure 5.13: ESD for the use case “move EV”

The sequences of interactions in Figure 5.13 can be summarized as follows:

- The stored electrical energy (EE) in the EV is moved to the EVP (i.e. transmit EE).
- The flow of EE is adjusted (i.e. regulate EE) in response to a control signal (C7) from the driver (i.e. transmit CS) which is converted from the driver input (DI2) (i.e. convert DI into CS).
- The adjusted EE is converted into torque (TQ2) at a linkage mechanism (i.e. convert EE into Torque).

- The torque (TQ2) of the linkage mechanism is transmitted to the rear axle (i.e. transmit torque) at the EV to move the EV (move EV), as illustrated in Figure 5.13.

The use case “power EV accessories” is linked to the actors “EV accessories” and “EV”, as shown in Figure 5.11. Figure 5.14 shows an enhanced sequence diagram for the use case “power EV accessories”.

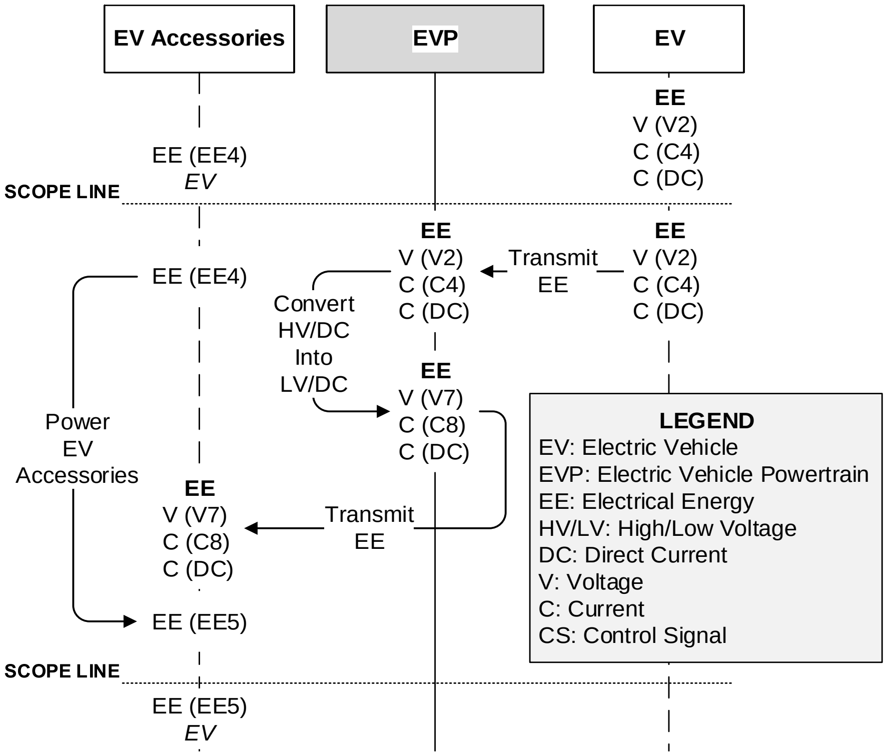


Figure 5.14: ESD for the use case “power EV accessories”

As illustrated in Figure 5.14, the stored electrical energy (EE) in the EV is moved to the EVP (i.e. transmit EE) and converted into low voltage (LV) electrical energy (i.e. convert HV/DC into LV/DC). Low voltage electrical energy is used (i.e. transmit EE) to power EV accessories (i.e. power EV accessories).

The determination of attribute values of relevant actors (including the initial and the final attributes) on each sequence diagram is important in terms of ensuring the coherency between state transitions of corresponding SSFDs of the same system. For example, sequence diagram for the use case “charge EV” in Figure 5.12 shows that electrical energy is stored in the EV (i.e. final state) and this stored electrical energy (i.e. initial state) is used in the use cases “move EV” and “power EV accessories”, as shown in Figure 5.13 and Figure 5.14, respectively.

5.4.2.4 System State Flow Diagram for EVP

Figure 5.15 shows corresponding SSFDs of the EVP based on the sequence diagrams of the EVP use cases represented in the previous section.

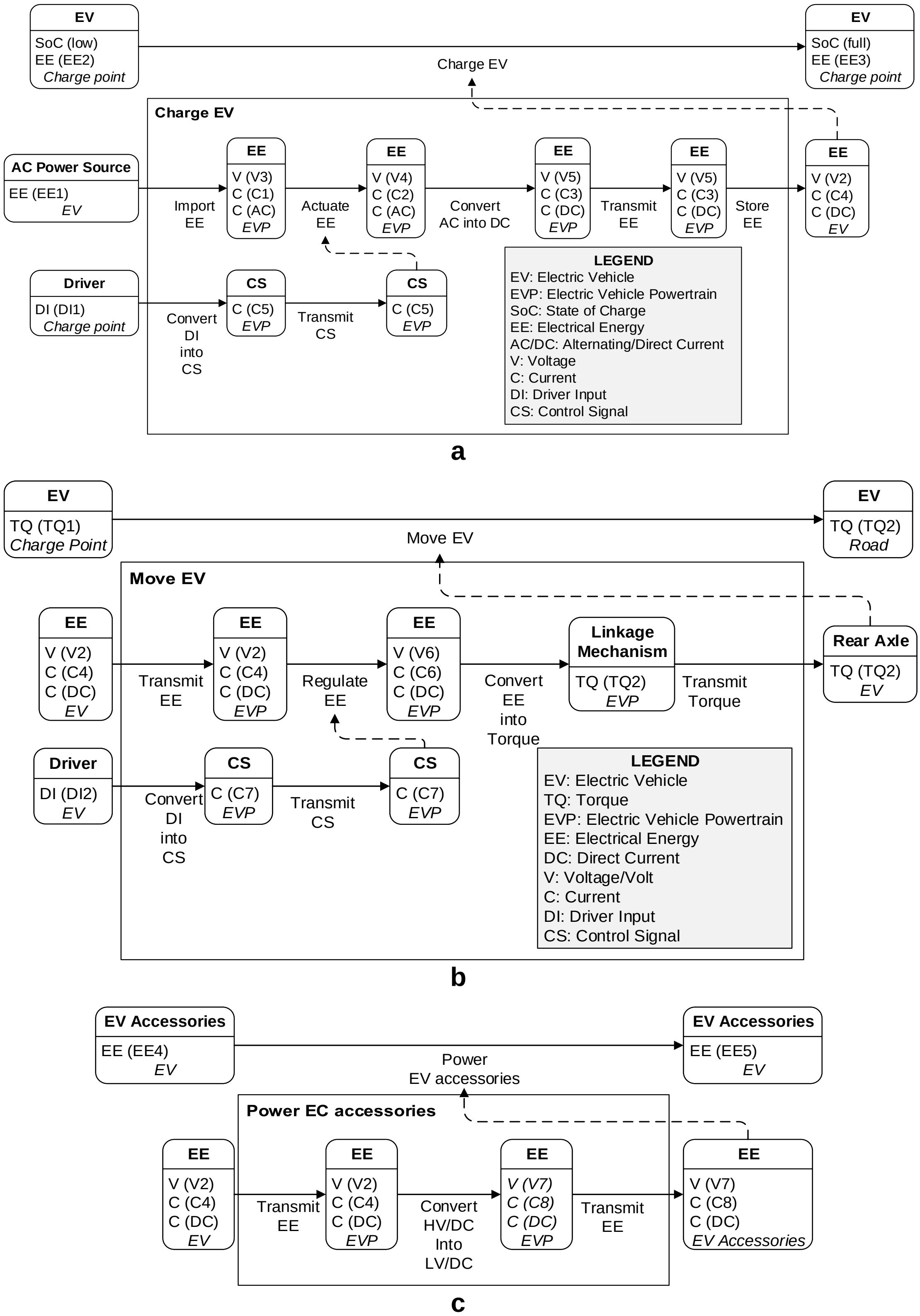


Figure 5.15: SSFDs for the use cases “charge EV” (a), “move EV” (b) and “power EV accessories” (c)

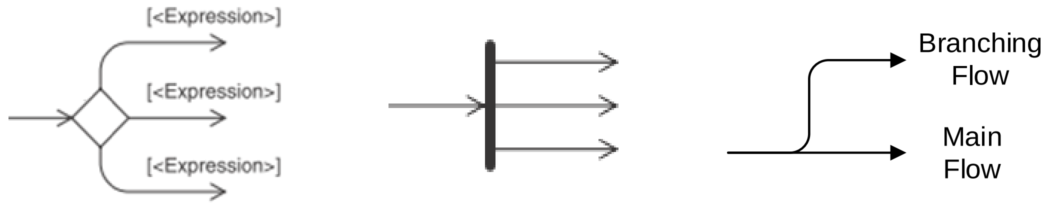
Two key points can be highlighted regarding the SSFDs in Figure 5.15:

- Connecting flows of the SSFDs for the use cases “charge EV” in Figure 5.15-a and “move EV” in Figure 5.15-b follow the same pattern, i.e. driver input (DI) is converted into a control signal (CS) which is linked to the flow of electrical energy (EE), however attributes of these connecting flows are different. Attribute values of the driver input are denoted by “DI1” and “DI2” on the SSFDs for the use cases “charge EV” and “move EV”, respectively. For the same use cases, current (C) values of the control signal are “C5” and “C7”, respectively. This shows that the use cases “charge EV” and “move EV” cannot take place simultaneously.

- The output state “electrical energy” of the SSFD in Figure 5.15-a is fed into the SSFDs in Figure 5.15-b and Figure 5.15-c and these SSFDs process this energy differently. SSFD for the use case “power EV accessories” in Figure 5.15-c converts this energy (V2) into lower voltage energy (V7) before transmitting it to the EV accessories, while SSFD for the use case “move EV” in Figure 5.15-b regulates the electrical energy (V6) in response to a control signal (C7) from the driver.

These key points show that the aggregation of the SSFDs in Figure 5.15 into a single diagram requires showing the same object with different attributes (e.g. control signal) in the same diagram, representing functional requirements of the EVP in a single SSFD. Therefore, relevant flow(s) should be divided into multiple flows for the representation of multiple functional requirements in the same diagram.

As mentioned in Section 5.2.1, SysML activity and state machine diagrams introduced the nodes “decision” and “choice”, respectively, to refer to different operation modes of a system by representing one incoming flow and several outgoing flows, which each have a condition. Both nodes are represented by the same notation, as shown in Figure 5.16-a. These diagrams also provide the node “fork” in the representation of flows which have one input flow and multiple output flows. Like the node “decision/choice”, the activity diagram and the state machine diagram of SysML represent the fork node by using the same notation, as illustrated in Figure 5.16-b. The focus of branching flow heuristic represented in Section 4.1.3 is to describe the flow of states (with functions) that branches out of the main flow by focusing on each object attribute change on the main flow. Figure 5.16-c shows the node of branching flow.



a-Decision/Choice node b-Fork node c-Branching flow node

Figure 5.16: Nodes "decision/choice" (a) and "fork" (b) in SysML activity and state machine diagrams and branching flow node (c) of the SSFD

Figure 5.16-a shows that the node "decision/choice" is limited to three output flows, while the node "fork" in Figure 5.16-b may have one input flow and multiple output flows. Each output flow on the fork node may be addressed independently or concurrently irrespective of any condition. The node of branching flow heuristic in Figure 5.16-c represents flows which take place concurrently. Therefore, a new notation is required for the representation of multiple functional requirements of a system in a single SSFD. The node "fork" in Figure 5.16-b is adapted to have output states with reference to an operation mode and optionally also with a parenthesized condition shown above the arrow. In SysML state machines (Friedenthal et al., 2008), the term "transition guard" is used to contain an expression that must be correct for the transition, while the term "condition" is used in the Statecharts (Harel, 1987). Figure 5.17 illustrates schema of the proposed fork node.

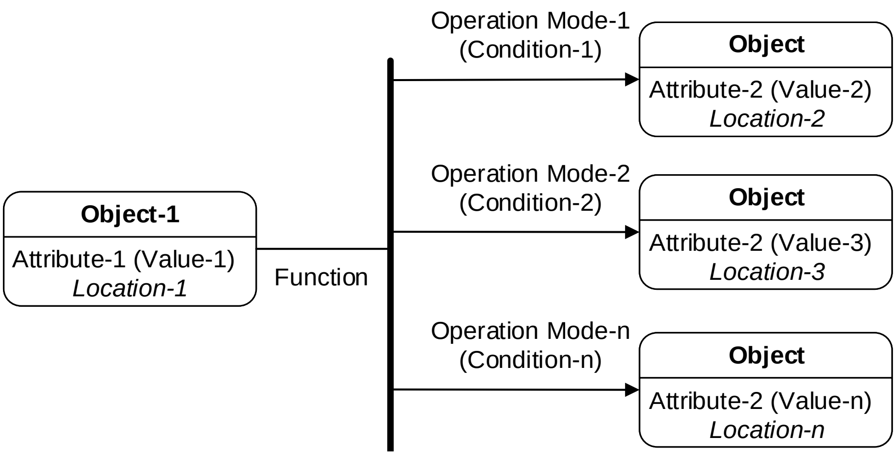


Figure 5.17: SSFD fork node

SysML fork node in Figure 5.16-b represents each output flow by separate arrows without describing any condition, while SSFD fork node divides the function arrow into multiple arrows, as illustrated in Figure 5.17. Part of the arrow between the input state (i.e. object-1) and the node represents the function text, while other part of the arrow can end up in multiple output states depending on the number of operation modes. By doing so, the same output object with different attribute values can be represented in respect of an input object with different attribute values.

Device-centric parts of the SSFDs for the use cases “charge EV”, “move EV” and “power EV accessories” in Figure 5.15 can be aggregated into a single SSFD by using the proposed fork node, as shown in Figure 5.18.

The use cases “charge EV”, “move EV” and “power EV accessories” are captured in a single SSFD by using two fork nodes, as shown in Figure 5.18.

- The first node shows that the flow of driver input (DI) follows one of two paths depending on its value. If the value is “DI1”, the input is converted into a control signal whose current value “C5” in respect of the function “actuate EE” of the use case “charge EV”. In the case of the value “DI2”, the control signal takes the value “C7” in related to the function “regulate EE” of the use case “move EV”.
- The second node represents the flow of electrical energy (EE) with respect to the operation modes “power EV accessories” and “move EV”.

The output state of the SSFD for the use case “charge EV” (i.e. electrical energy) is a shared state between the SSFDs in Figure 5.15. Location of this state is shown as “EVP” in Figure 5.18 for the sake of the aggregation of the EVP SSFDs.

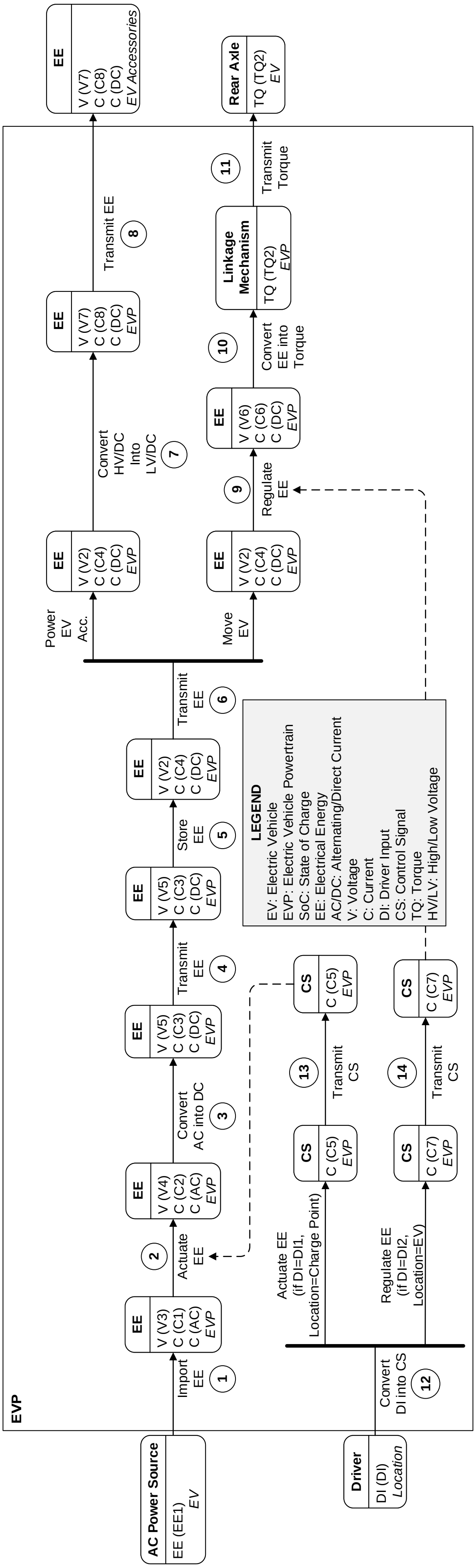


Figure 5.18: EVP SSFD

From the SSFD in Figure 5.18 a high level EVP function tree can be extracted, as shown in Figure 5.19.

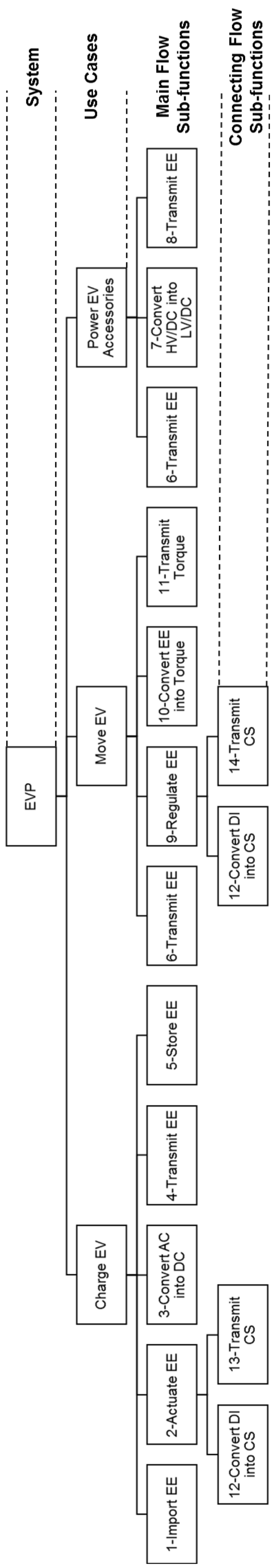


Figure 5.19: EVP function tree

Figure 5.19 represents the EVP functions numbered in Figure 5.18 at three levels:

- The first level shows the analysed use cases in respect to the main functional requirements of the EVP;
- The second level categorizes the main flow functions in Figure 5.18 into these use cases;
- The third level shows the connected flow functions which are required by the main flow functions “actuate EE” and “regulate EE” of the use cases “charge EV” and “move EV”, respectively.

As shown in Figure 5.19, some use cases possess the same function with the same number.

- The function “transmit EE” numbered as “6” is shared by the use cases “move EV” and “power EV accessories”. This function is a transmission function, that is, it changes the object location. The input EE location and the output EE location of this function for both use cases are shown as “EVP” in Figure 5.18 due to the fact that the EVP is to be designed.
- The function “convert DI into CS” numbered as “12” is shared by the use cases “charge EV” and “move EV”. This function is a conversion function, that is, it modifies the object attributes (as well as the object location is some cases). The input DI attributes and the output CS attributes of this function for both use cases are different, as shown in Figure 5.18.

From Figure 5.19 we can also see that some use cases possess the same function with different number. For example, the function “transmit CS” numbered as “13” and “14” for the use cases “charge EV” and “move EV”, respectively. The output CS attributes of these functions are different for both use cases. The output CS of the number “13” is linked to the input state of the function “actuate EE”, while the output CS of the number “14” is related to the input state of the function “regulate EE”. Table 5.1 shows object attribute relations for each numbered function in Figure 5.18 and Figure 5.19.

Table 5.1: Object attribute relations in the EVP SSFD

Function Number	Relations
1	EE (V3,C1,AC,EVP) = f (AC Power Source (EE1,EV))
2	EE (V4,C2,AC,EVP) = f (EE (V3,C1,AC,EVP).CS(C5,EVP))
3	EE (V5,C3,DC,EVP) = f (EE (V4,C2,AC,EVP))
4	EE (V5,C3,DC,EVP) = f (EE (V5,C3,DC,EVP)
5	EE (V2,C4,DC,EVP) = f (EE (V5,C3,DC,EVP)
6	*Power EV Acc.:* EE (V2,C4,DC,EVP) = f (EE (V2,C4,DC,EVP)
	Drive EV: EE (V2,C4,DC,EVP) = f (EE (V2,C4,DC,EVP)
7	EE (V7,C8,DC,EVP) = f (EE (V2,C4,DC,EVP)
8	EE (V7,C8,DC,EV Accessories) = f (EE (V7,C8,DC,EVP)
9	EE (V6,C6,DC,EVP) = f (EE (V2,C4,DC,EVP).CS(C7,EVP))
10	Linkage Mechanism (TQ2,EVP) = f (EE (V6,C6,DC,EVP)
11	Rear Axle (TQ2,EV) = f(Linkage Mechanism (TQ2,EVP))
12	*Actuate EE:* CS (C5,EVP) = f (Driver (DI1, Charge Point))
	Regulate EE: CS (C7,EVP) = f (Driver (DI2, EV))
13	CS (C5,EVP) = f(CS (C5,EVP))
14	CS (C7,EVP) = f(CS (C7,EVP))

As with the SSFD in Figure 5.18, Table 5.1 shows the majority of the object location attributes as “EVP”. Because, the EVP SSFD was developed in a solution neutral way (as discussed before in related to Figure 5.19). For example, for the function “import EE” numbered as “1” in Figure 5.18, the output location of electrical energy (EE) can be specified as “cable” once design decisions are made on the EVP. Object attribute relations for the functions “transmit EE” numbered as “6” and “convert DI into CS” numbered as “12” in Table 5.1 are shown in two rows with respect to different operation modes.

5.4.2.5 EVP Case Study - summary of key points

This section introduced the development of the SSFD for a system with multiple operation modes on the basis of the analysis of an electric powertrain vehicle (EVP). The development of the SSFD was carried based on the following stages:

1) **Use Case Diagram:** The first stage represented some of the high level functionality of the EVP in terms of a use diagram.

2) **Enhanced Sequence Diagram (ESD):** The second stage was about developing enhanced sequence diagrams for the EVP use cases related to its main functional requirements described in Section 5.4.2.1.

3) **System State Flow Diagram:** This stage aimed to represent corresponding SSFDs based on the ESDs.

4) **Aggregation of the SSFDs into a single diagram:** Unlike the FSVC, the EVP has multiple functional requirements and this required to develop a SSFD for each functional requirement using the ESD. This section introduced a new notation, called the SSFD fork node, with the aim of combining different SSFDs into a single diagram.

The development of the function tree based on the EVP SSFD highlighted the role of the functions in respect of the EVP use cases. The formulation of object attribute relations for each EVP function provided a detailed view about the SSFD function model.

5.4.3 Active Rear Spoiler

5.4.3.1 Case study background

The three step process of developing a SSFD represented in Section 5.3.1 was based on the development of one SSFD in respect of one functional requirement. The previous section added one more step (i.e. aggregation of the SSFDs using the SSFD fork node) for the deployment of this process to develop a system with multiple operation modes. This section seeks to test the validity of the same process summarized in Section 5.4.2.5 on an active rear spoiler (ARS). The analysis of the ARS, applicable to sports utility vehicles (SUV), will be based on design specifications came from a major automotive company. According to the specifications from the company, the spoiler has certain amount of autonomy with user activated controls and four modes of operation, as shown in Table 5.2.

Table 5.2: Four modes of operation of an active rear spoiler

FOUR MODES OF OPERATION			
SHOWROOM & PARKED	**AERO MODE Normal Speed**	**AERO MODE High Speed**	**SOILING MODE**
• Desirable exterior aesthetic when in stowed state	• Low drag • ~10-12 degs • Trailing edge movement in X by 100mm • Extension of spoiler to enable drag savings	• Reduced spoiler angle: ~30degs • Spoiler produces downforce/negative lift as required. Will increase drag in this state	• Rear end soiling in poor road soiling conditions • 15-25mm slot width • Centre section open • Reduced rear end soiling, without a drag / CO_2 detriment • Slots will increase drag

This section focuses on the reduction of aerodynamic drag on a vehicle by addressing relevant modes of the spoiler. Therefore, soiling state and the reduction of rear vehicle lift will not be considered here. Figure 5.20 specifies modes of operation of the spoiler which will be the focus of this section.

MODES OF OPERATION
Showroom & Parked: $0 \leq VV = VV1$; $0 \leq D = D1$; $\theta = 0^{o}$
Normal Speed: $VV1 < VV = VV2$; $D1 < D = D2$; $\theta = 10^{o}$
High Speed: $VV2 < VV = VV3$; $D2 < D = D3$; $\theta = 30^{o}$
VV: Vehicle Velocity; D: Drag Count; Θ: Spoiler Angle

Figure 5.20: Modes of operation of the spoiler with specific attribute values

The status of the vehicle is shown in terms of modes of operation "showroom & parked", "normal speed" and "high speed" in Figure 5.20. As shown in Figure 5.20, the velocity value and the drag count value of the vehicle at "showroom & parked" state may be "0" or "V1" and "D1", respectively.

Arguably, the spoiler can operate by following operation modes in Table 5.2 from left to right. However, depending on the environment and driver preference, the flow of modes may be different. Therefore, this section analyses operation modes of the spoiler in Figure 5.20 based on the assumption that the flow of spoiler state will follow the order in Figure 5.21.

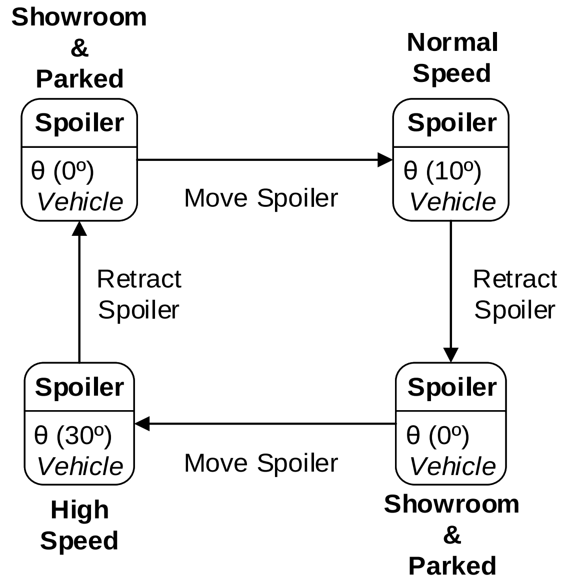


Figure 5.21: The flow of spoiler states in respect of the vehicle status

Figure 5.21 shows the flow of spoiler state in terms of SSFD state transitions. The spoiler controls the main flow of material (i.e. air) through the vehicle by changing the global attribute “spoiler angle” with respect to the operation modes “showroom & parked”, “normal speed” and “high speed, as shown in Figure 5.21.

5.4.3.2 Use Case Diagram for ARS

Figure 5.22 shows a use case diagram of the ARS based on operation modes of the spoiler in Figure 5.21.

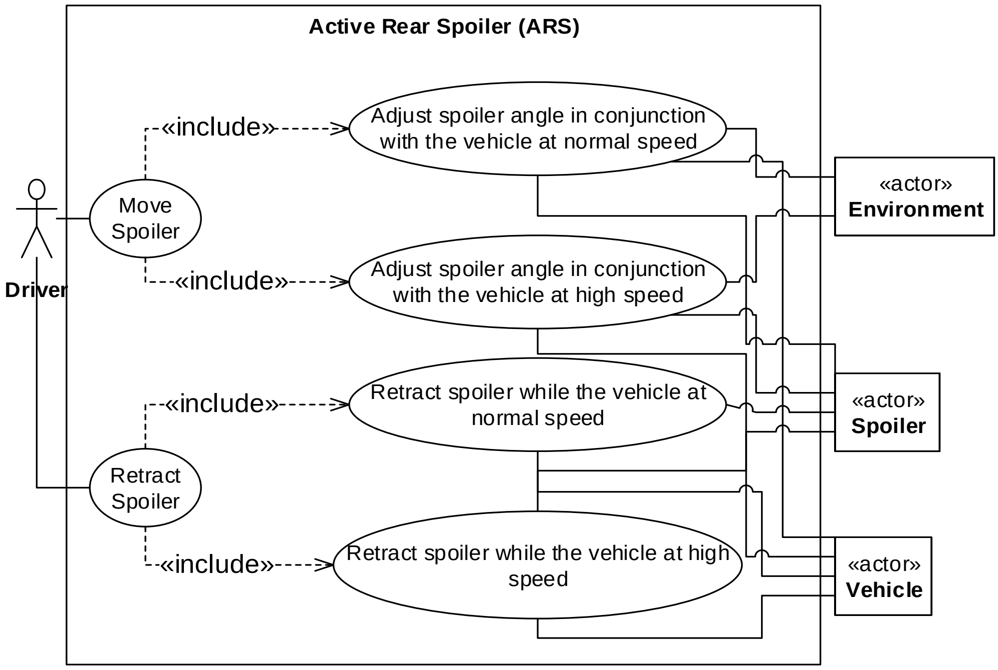


Figure 5.22: Use case diagram of the ARS

As shown in Figure 5.22, the use case “move spoiler” and “retract spoiler” include two sub-use cases related to the spoiler operation modes represented in Figure 5.21.

5.4.3.3 Enhanced Sequence Diagram for ARS

The operation modes “aero mode-normal speed” and “aero mode-high speed” of the spoiler in Table 5.2 are addressed through the use cases “adjust spoiler angle in conjunction with the vehicle at normal speed” and “adjust spoiler angle in conjunction with the vehicle at high speed” in Figure 5.22. These use cases require the actors “driver”, “environment (i.e. air)”, “spoiler” and “vehicle”. Figure 5.23 shows an enhanced sequence diagram for the use case “adjust spoiler angle in conjunction with the vehicle at normal speed”.

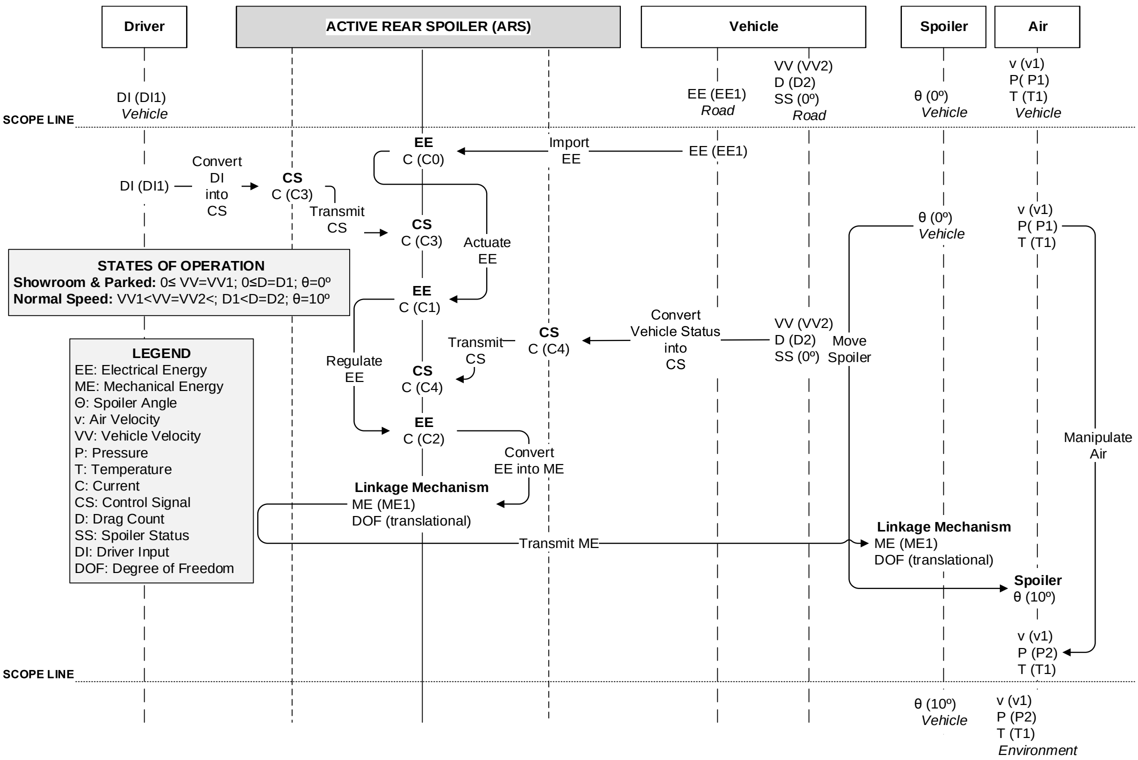


Figure 5.23: ESD for the use case “adjust spoiler angle in conjunction with the vehicle at normal speed”

The sequences of interactions in Figure 5.23 can be summarized as follows:

- Electrical energy (EE1) is provided from the vehicle (i.e. import EE).
- The flow of imported electrical energy is commenced (i.e. actuate EE) in response to a control signal (C3) (i.e. transmit CS) which is converted from the driver input (DI1) (i.e. convert DI into CS).

- The flow of EE is adjusted (i.e. regulate EE) in response to a control signal (C4) (i.e. transmit CS) which is generated based on the vehicle status (i.e. convert vehicle status into CS).

- The adjusted EE is converted into mechanical energy (ME1) at a linkage mechanism (i.e. convert EE into ME) and transmitted to the spoiler (i.e. transmit ME).

- The transmitted ME (ME1) changes the position of the spoiler (i.e. move spoiler).

- The drag on the vehicle reduces as a result of the change of the spoiler position (i.e. manipulate air), as illustrated in Figure 5.23.

Sequence diagram for the use case "adjust spoiler angle in conjunction with the vehicle at high speed" follows the same pattern with different attribute values (see Appendix A for detail).

The actors "driver", "vehicle" and "spoiler" are related to the use cases "retract spoiler while the vehicle at normal speed" and "retract spoiler while the vehicle at high speed" in Figure 5.22. Figure 5.24 illustrates a sequence diagram for the former.

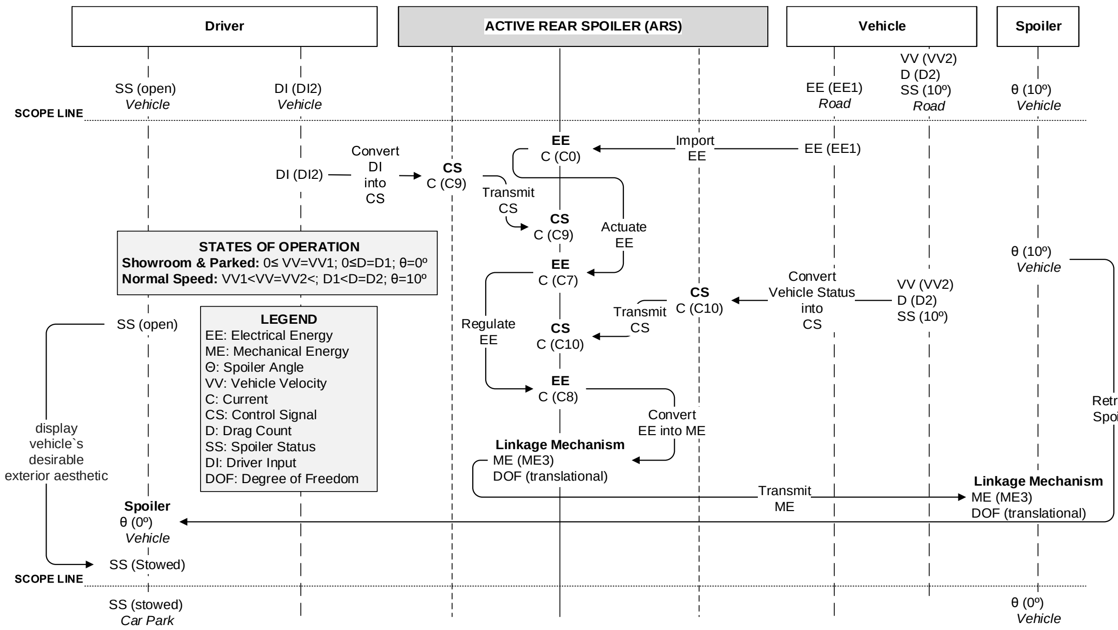


Figure 5.24: ESD for the use case "retract spoiler while the vehicle at normal speed"

Sequence diagrams in Figure 5.23 and Figure 5.24 are alike except that two features of the diagrams are different. The environment (i.e. air) is not required by the use case “retract spoiler while the vehicle at normal speed” and most of the actors possess different attribute values than the ones in Figure 5.23. The second difference is that the intended effect of the ARS is related to the driver in Figure 5.24 instead of the air. The retracted spoiler affects the status of the driver, as shown in Figure 5.24.

The pattern of the sequence diagram in Figure 5.24 is followed by the use case “retract spoiler while the vehicle at high speed” with different attribute values (see Appendix B for detail).

5.4.3.4 System State Flow Diagram for ARS

By following the same methodology explained on the FSVC in Section 5.3.2.3 and further exemplified on the EVP in Section 5.4.2.4, SSFD for the use case “adjust spoiler angle in conjunction with the vehicle at normal speed” can be represented based on its sequence diagram in Figure 5.23. Figure 5.25 shows the SSFD for this use case.

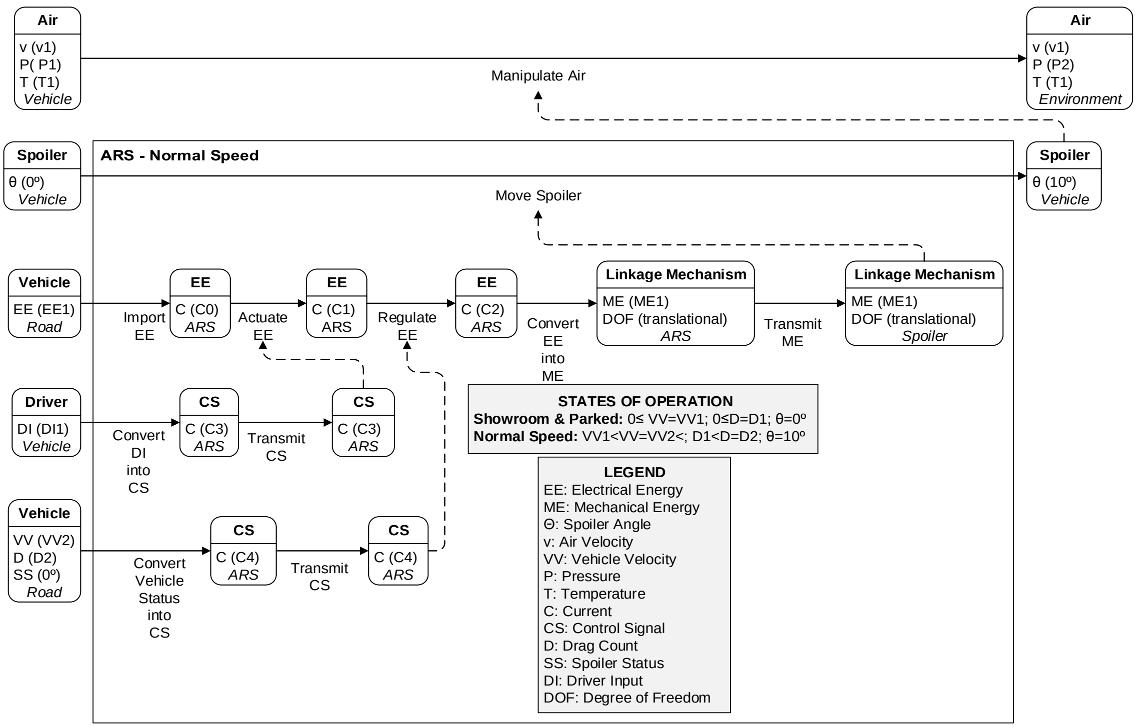


Figure 5.25: SSFD for the use case “adjust spoiler angle in conjunction with the vehicle at normal speed”

SSFD for the use case “adjust spoiler angle in conjunction with the vehicle at high speed” will be the same as Figure 5.25 except that the attribute values will be different (see Appendix C for detail).

Similarly, SSFD for the use case “retract spoiler while the vehicle at normal speed” can be represented based on its sequence diagram in Figure 5.24, as shown in Figure 5.26.

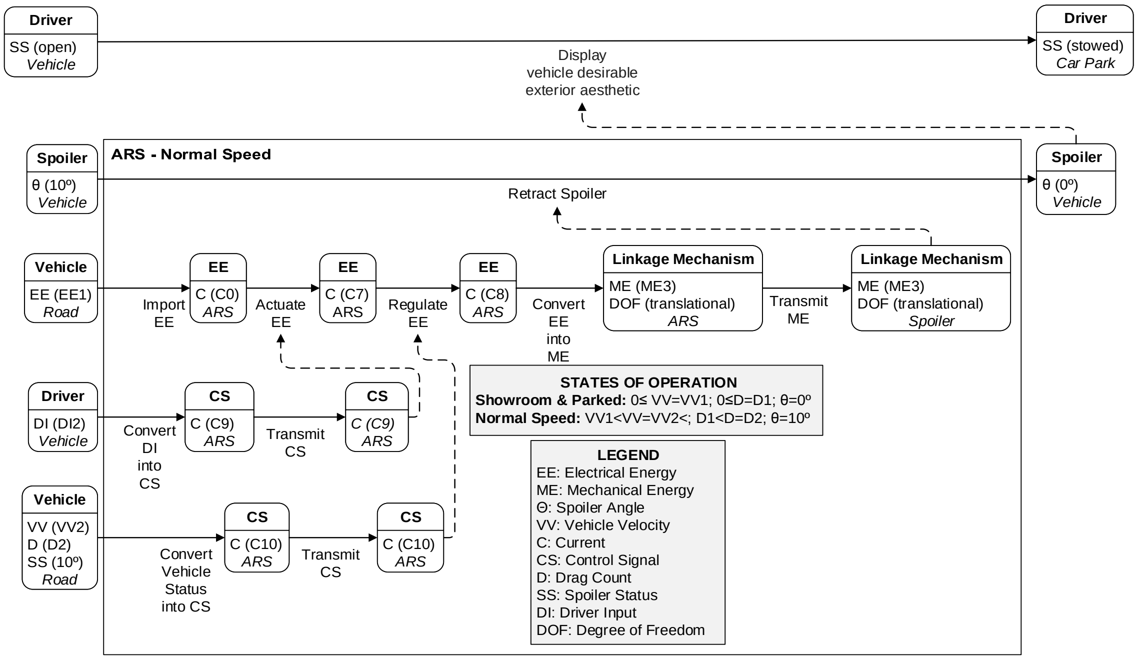


Figure 5.26: SSFD for the use case “retract spoiler while the vehicle at normal speed”

SSFD for the use case “retract spoiler while the vehicle at high speed” will be based on the same structure as Figure 5.26, but it will possess different attribute values (see Appendix D for detail).

Device-centric views of the developed SSFDs for the ARS sub-use cases can be aggregated into a single diagram by exploiting the SSFD fork node described in Section 5.4.2.4. Figure 5.27 shows SSFD for the ARS.

There are two key points regarding the diagram in Figure 5.27:

- The output state of the function “move spoiler” is a shared state between the SSFDs of the ARS use cases. Location of this state is shown as “ARS” in Figure 5.27 for the sake of the aggregation of these SSFDs.

- As with the FSVC SSFD in Figure 5.10 and the EVP SSFD in Figure 5.18, Figure 5.27 shows that the majority of the object location attributes are denoted as the name of the system per se (i.e. ARS) due to the fact that the system is to be designed.

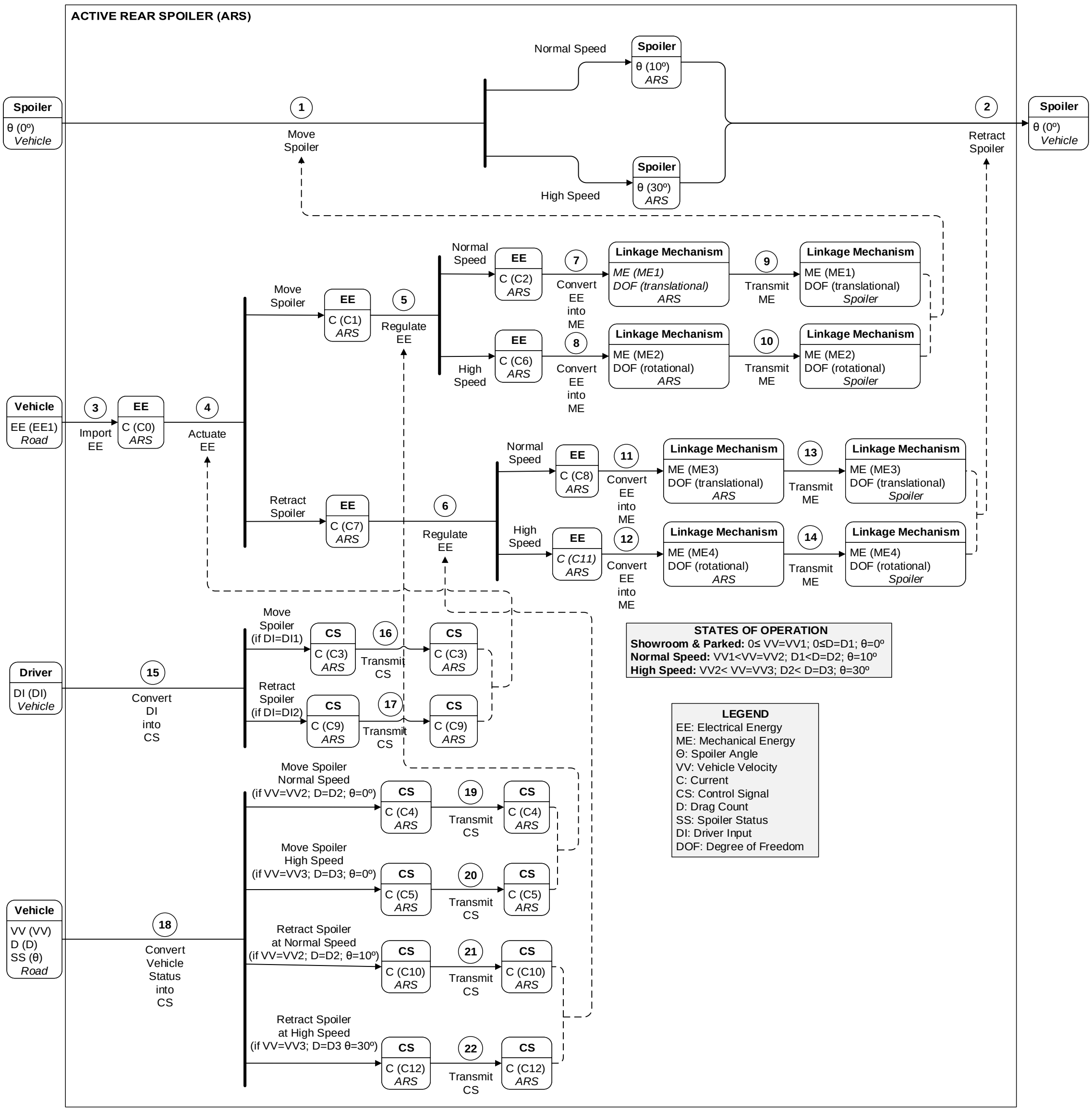


Figure 5.27: ARS SSFD

Figure 5.27 includes 6 fork nodes with respect to the operation modes of the ARS. Figure 5.28 represents the node related to the state of spoiler as an excerpt from the diagram in Figure 5.27.

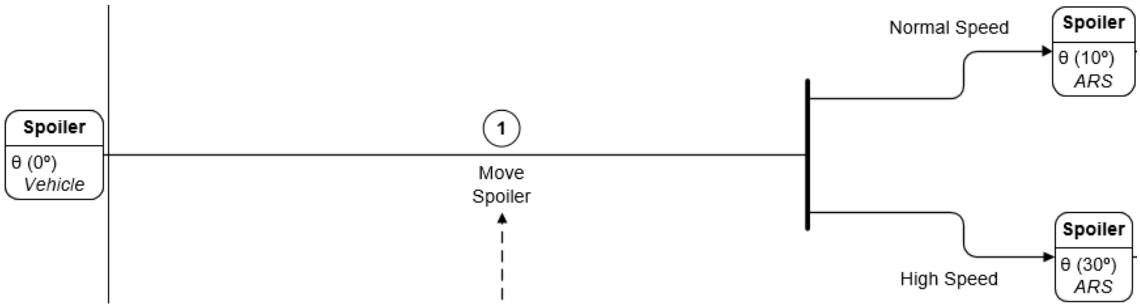


Figure 5.28: The fork node in respect of the state of spoiler

Figure 5.28 shows that the flow of spoiler follows the path of “normal speed” or “high speed” in response to the flow of mechanical energy (see Figure 5.27 for the flow of mechanical energy). Figure 5.29 shows the node related to the flow electrical energy (EE) as an excerpt from the diagram in Figure 5.27.

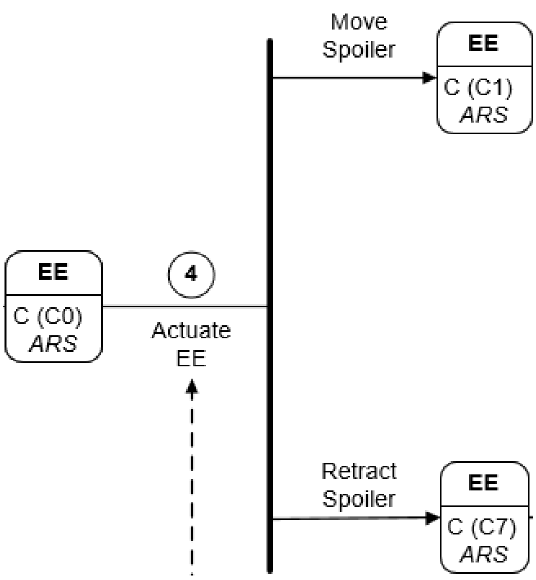
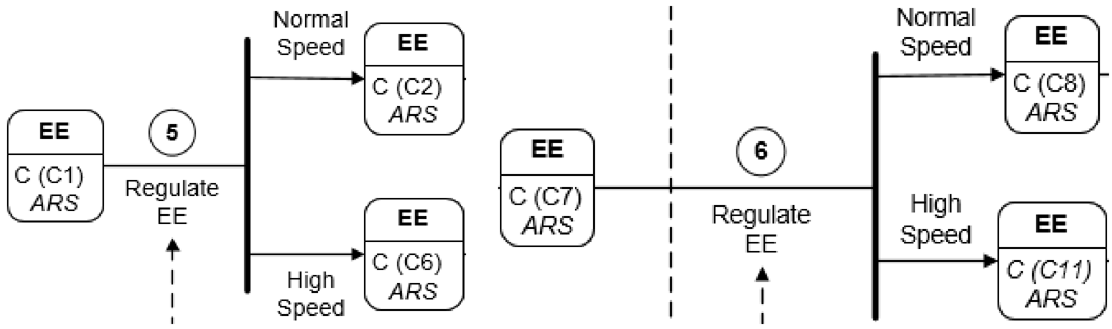


Figure 5.29: The fork node in respect of the flow electrical energy (EE)

As shown in Figure 5.29, the flow of electrical energy (EE) follows the path of “move spoiler” or “retract spoiler” in response to the flow of control signal converted from driver input (see Figure 5.27 for the flow of control signal). Figure 5.30 shows another node related to the flow electrical energy as an excerpt from the diagram in Figure 5.27.



a b

Figure 5.30: The fork node in respect of the flow of electrical energy (EE)

Figure 5.30 points out that the flow of electrical energy follows the path of "normal speed" or "high speed" in response to the flow of control signal converted from vehicle status (see Figure 5.27 for the flow of control signal). Figure 5.30-a and Figure 5.30-b are associated with the operation modes "move spoiler" and "retract spoiler", respectively. Figure 5.31 illustrates the node related to the flow of driver input (DI) as an excerpt from the diagram in Figure 5.27.

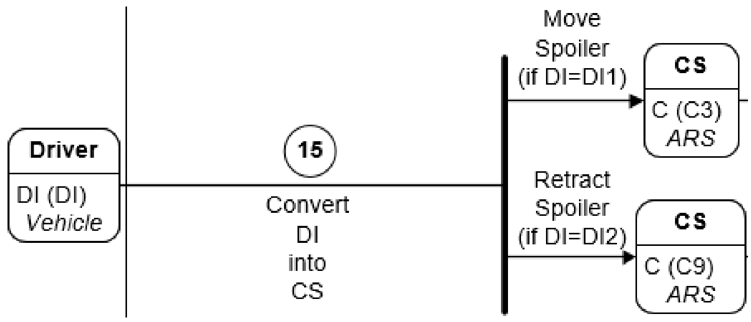


Figure 5.31: The fork node in respect of the flow of driver input (DI)

The flow of driver input follows the path of "move spoiler" or "retract spoiler" depending on the value of driver input (DI), as shown in Figure 5.31. Figure 5.32 depicts the last fork node on the SSFD diagram in Figure 5.27 as an excerpt.

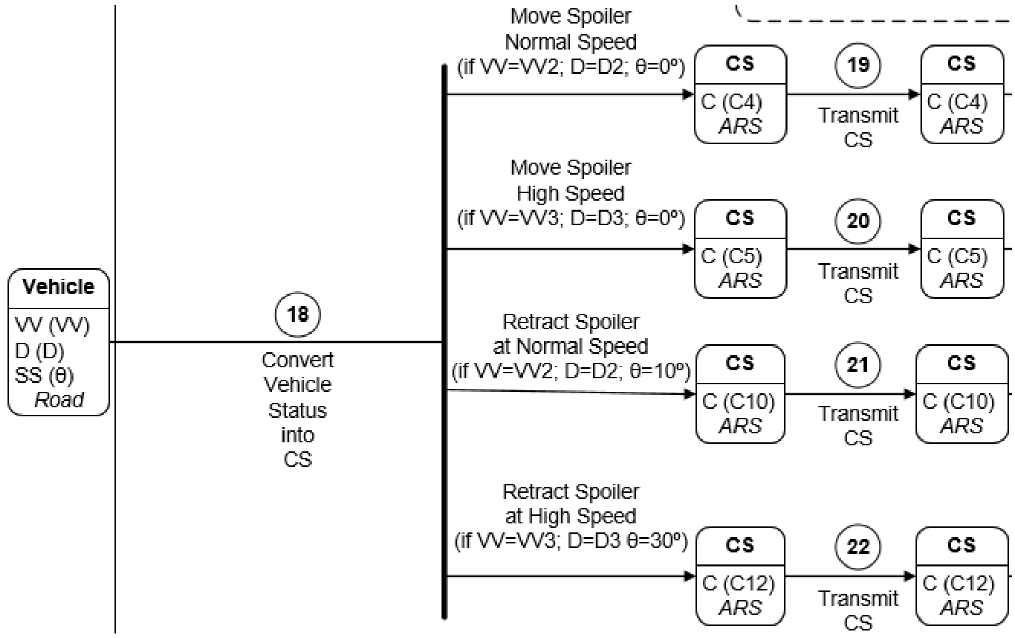


Figure 5.32: The fork node in respect of the flow of control signal (CS)

Figure 5.32 shows that the flow of control signal (CS) converted from vehicle status addresses all operation modes of the ARS. The path of the flow depends on the value of vehicle speed (V), drag count (D) and spoiler angle (θ).

From the SSFD in Figure 5.27 a high level ARS function tree can be extracted, as shown in Figure 5.33. The tree represents the ARS functions vertically for maximum legibility.

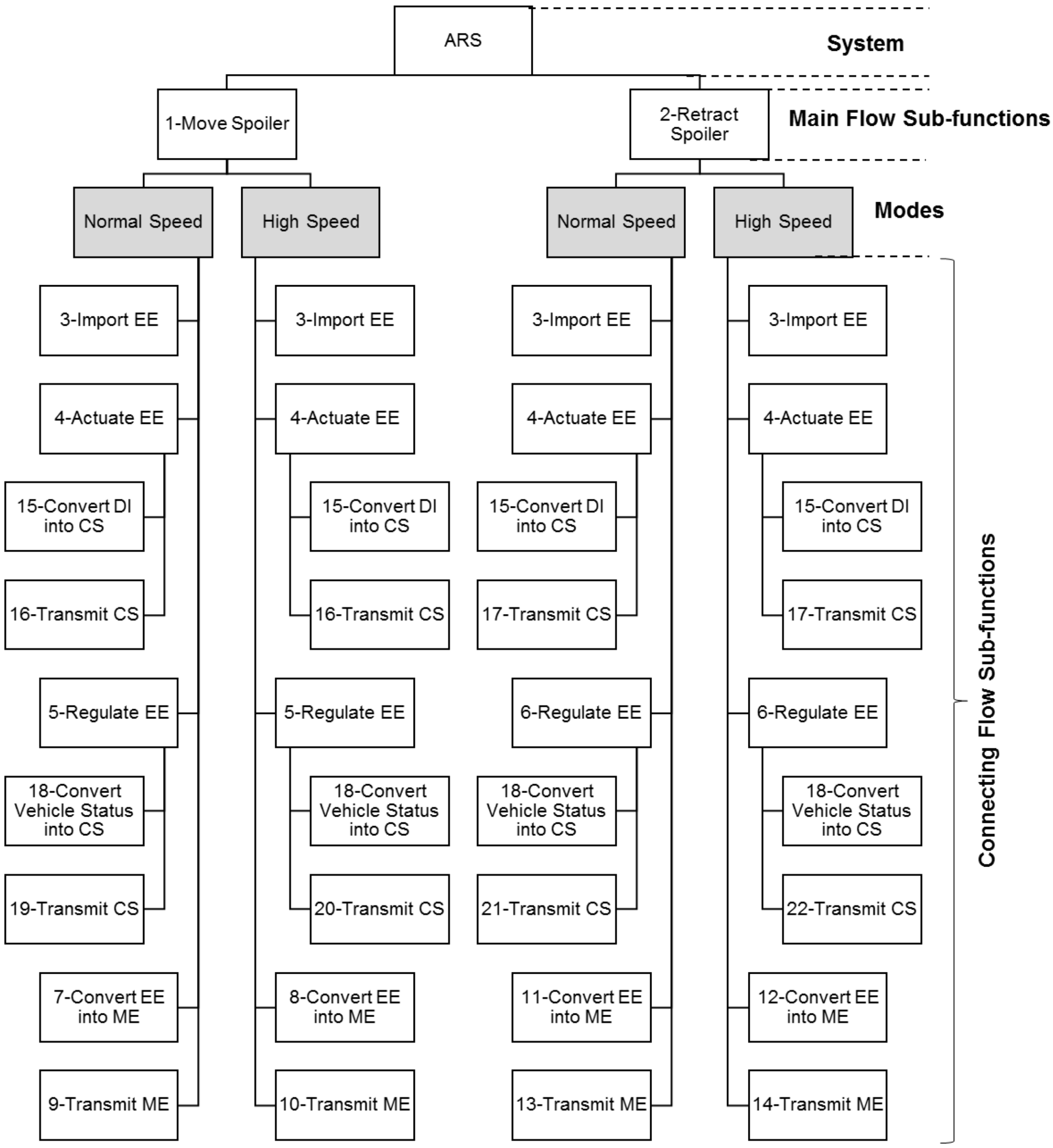


Figure 5.33: ARS function tree

Figure 5.33 represents the ARS functions numbered in Figure 5.27 at two levels. The first level shows the main flow functions of the ARS. These functions are also shown as the main use cases of the ARS in Figure 5.22. The second level represents the connected flow functions based on relevant operation mode of the spoiler which is shown in grey. The mode “showroom & parked” is included in the operation modes “normal speed” and “high speed”. It is important to note that the main functions “move spoiler” and “retract spoiler” include the same connecting flow functions with the same number (e.g. the function “actuate EE” numbered as “4”) and with the different number (e.g. the function “transmit ME” numbered as “9”, “10”, “13” and “14”). Table 5.3 shows object attribute relations for each numbered function in Figure 5.27 and Figure 5.33.

Table 5.3: Object attribute relations in the ARS SSFD

Function Number	Relations
1	*Normal speed:* Spoiler (10º, ARS) = f (Spoiler (0º, Vehicle). Linkage Mechanism (ME1,Translational,Spoiler))
	High speed: Spoiler (30º, ARS) = f (Spoiler (0º, Vehicle). Linkage Mechanism (ME2,Rotational,Spoiler))
2	*Normal speed:* Spoiler (0º, Vehicle) = f (Spoiler (10º, ARS). Linkage Mechanism (ME3,Translational,Spoiler))
	High speed: Spoiler (0º, Vehicle) = f (Spoiler (30º, ARS). Linkage Mechanism (ME4, Rotational, Spoiler))
3	EE (C0,ARS) = f (Vehicle (EE1,Road))
4	*Move Spoiler:* EE (C1,ARS) = f (EE (C0,ARS).CS(C3,ARS))
	Retract Spoiler: EE (C7,ARS) = f (EE (C0,ARS).CS(C9,ARS))
5	*Normal Speed:* EE (C2,ARS) = f (EE (C1,ARS).CS(C4,ARS))
	High Speed: EE (C6,ARS) = f (EE (C1,ARS).CS(C5,ARS))
6	*Normal Speed:* EE (C8,ARS) = f (EE (C7,ARS).CS(C10,ARS))
	High Speed: EE (C11,ARS) = f (EE (C7,ARS).CS(C12,ARS))
7	Linkage Mechanism (ME1,Translational,ARS) = f(EE(C2,ARS))
8	Linkage Mechanism (ME2,Rotational,ARS) = f(EE(C6,ARS))
9	Linkage Mechanism (ME1,Translational,Spoiler) = f(Linkage Mechanism (ME1,Translational,ARS))
10	Linkage Mechanism (ME2,Rotational,Spoiler) = f(Linkage Mechanism (ME2,Rotational,ARS))
11	Linkage Mechanism (ME3,Translational,ARS) = f(EE(C8,ARS))
12	Linkage Mechanism (ME4,Rotational,ARS) = f(EE(C11,ARS))
13	Linkage Mechanism (ME3,Translational,Spoiler) = f(Linkage Mechanism (ME3,Translational,ARS))
14	Linkage Mechanism (ME4,Rotational,Spoiler) = f(Linkage Mechanism (ME4,Rotational,ARS))
15	*Move Spoiler:* CS (C3,ARS) = f(Driver(DI1,Vehicle)
	Retract Spoiler: CS (C9,ARS) = f(Driver(DI2,Vehicle)
16	CS (C3,ARS) = f(CS (C3,ARS))
17	CS (C9,ARS) = f(CS (C9,ARS))
18	*Move Spoiler/Normal Speed:* CS (C4,ARS) = f(Vehicle(VV2,D2,0º,Road)
	Move Spoiler/High Speed: CS (C5,ARS) = f(Vehicle(VV3,D3,0º,Road)
	Retract Spoiler at Normal Speed: CS (C10,ARS) = f(Vehicle(VV2,D2,10º,Road)
	Retract Spoiler at High Speed: CS (C12,ARS) = f(Vehicle(VV3,D3,30º,Road)
19	CS (C4,ARS) = f(CS (C4,ARS))
20	CS (C5,ARS) = f(CS (C5,ARS))
21	CS (C10,ARS) = f(CS (C10,ARS))
22	CS (C12,ARS) = f(CS (C12,ARS))

5.4.3.5 ARS Case Study - summary of key points

This section introduced a different example of the development of a SSFD for a system with multiple operation modes. The main difference of the ARS to the EVP is to address multiple operation modes by changing the global attribute “location” of the spoiler rather than modifying its local attributes. Like the development of the FSVC SSFD and the EVP SSFD, the first step was to representation of a use diagram in respect of the functionality of the ARS. This followed the development of an ESD for each ARS sub-use case. Next, corresponding SSFDs were represented based on these ESDs and they were combined into a single diagram by means of the proposed SSFD fork node.

The ARS function tree in Figure 5.33 highlighted the same functions with the same number and different number on the ARS SSFD in Figure 5.27, while Table 5.3 detailed object attribute relations for each numbered function on the ARS SSFD and function tree.

5.5 Chapter summary

This chapter represented an extended version of the SSFD framework for function modelling of systems with multiple operation modes by introducing "Enhanced Sequence Diagram (ESD)" and "SSFD fork node". The proposed enhanced sequence diagram describes and maps individual flows of material, energy and information associated with the timeline of an actor. The introduced methodology for the development of the SSFD based on the "Enhanced Sequence Diagram" enhances the ability of the SSFD to represent flows through a system in a coherent way. The fork node supports the integration of multiple SSFDs of the same system into a single diagram by enabling the representation of conditional flows. The extended framework was represented as a four step process as follows:

1) Represent the functionality of the system by a Use Case Diagram,
2) Represent sequences of states and functions through the system for each use case (operation mode) through an Enhanced Sequence Diagram (ESD),
3) Represent corresponding SSFD for each use case through the ESD,
4) Aggregate the SSFDs of the use cases into a single diagram through the proposed SSFD fork node(s).

The concept of developing a SSFD through an ESD based on one use case (the first three steps of the process) was illustrated on a front split view camera. The case studies of electric vehicle powertrain and an active rear spoiler focused on the modelling of multiple operation modes by using the SSFD fork node(s) in the integration of SSFDs into a single diagram.

6. Discussion

6.1 Summary of research methodology

The main aim of this thesis was to develop the SSFD methodology for function modelling of complex multidisciplinary systems and validate it through case studies. As discussed in Section 1.4, the research methodology followed for the achievement of this aim was based on an iterative mapping between induction and deduction. Figure 6.1 shows the key elements of the thesis chapters based on the research methodology presented in Figure 1.2.

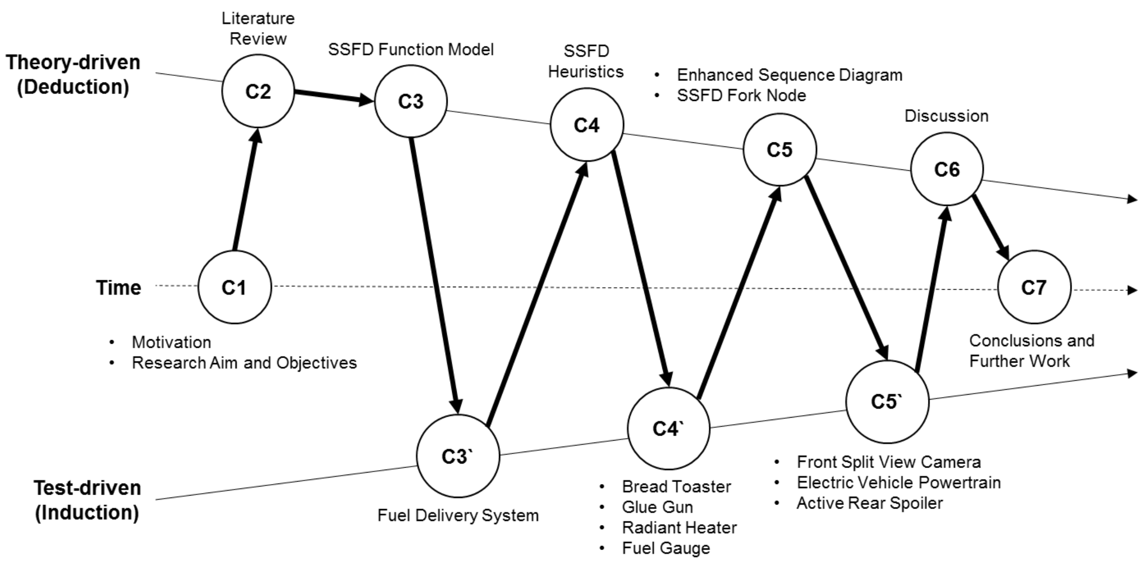


Figure 6.1: A revised version of research methodology

The key points of the methodology in Figure 6.1 are summarized as follows:

- The proposed approach in this thesis was based on the System State Flow Diagram (SSFD), which has previously been introduced (see Campean and Henshall, 2008) in a simplistic fashion. The developed “SSFD function model” aimed to support the SSFD creation and elicitation aspects.
- The deployment of the SSFD function model to represent the given function chains of an in-tank fuel delivery system demonstrated the need for a set of steps, called “SSFD heuristics”, for the establishment of functional model of a system using the SSFD function model.
- The SSFD heuristics were illustrated on a “bread toaster”. The validation of the SSFD heuristics on the desktop case studies “glue gun”, “radiant

heater" and "fuel gauge" showed a need for the development of the SSFD framework for the analysis of systems with multiple operation modes.

- This need resulted in the introduction of "Enhanced Sequence Diagram (ESD)" for the development of a SSFD by taking into account sequences of states and functions through the system. The ESD was illustrated based on one operation mode of the industrial case study "front view split camera (FSVC)" as part of a three step process for the development of the SSFD from the ESD. The "SSFD fork node" was proposed for the aggregation of different SSFDs of the same system into a single diagram. The developed SSFD framework including the ESD and the SSFD fork node was validated on the "electric vehicle powertrain (EVP)" and the "active rear spoiler (ARS)" industrial case studies.

This chapter presents a critical review of the methodology taken to conduct this research in the context of the key developments introduced to address the limitations of the current SSFD represented in Section 1.2. The review is presented based on three sections:

- Section 6.2 discusses "SSFD Function Model" and "SSFD Heuristics", which are the key concepts of the SSFD framework, in function modelling with a strong focus on the work in Chapter 4 in which the deployment of the framework to develop function modelling of systems with one mode of operation was represented.
    - Section 6.2.1 summarizes similarities and differences between the SSFD function model and other established functional modelling frameworks. The section concludes with a critical review of the experience of using the SSFD function model in function modelling of the case studies.
    - Section 6.2.2 compares the SSFD heuristics with the approaches of Pahl et. al (2007) and Otto and Wood (2001). This follows a critical review of the experience of using the SSFD heuristics in function modelling of the case studies.
    - Section 6.2.3 discusses the use of the SSFD framework in the representation of expected and structure behaviour of a system.

The section shows how the SSFD framework supports consistent integration of multiple levels of analysis through a nested system. The bread toaster case study is used as an illustrative example.

- Section 6.3 focuses on the work in Chapter 5 and discusses the use of the SSFD framework in function modelling of complex multidisciplinary systems with multiple operation modes. Section 6.3.1 reflects on the "complex multidisciplinary" claim, while Section 6.3.2 argues for the case for function modelling of systems with multiple operation modes. The section concludes with a summary of the contribution of the SSFD framework in relation to the weaknesses of the current function modelling approaches discussed in Section 2.5.
- Section 6.4 demonstrates the effectiveness of the SSFD framework on supporting systems engineering design, in particular to failure mode avoidance process (see Section 6.4.1), systems engineering design integration and engineering change management (see Section 6.4.2).

6.2 The SSFD framework in function modelling

This research introduced the SSFD framework to address the shortcomings of the current function modelling approaches used in practice. The necessity of developing a framework for the SSFD was based on the need to the development of function modelling of a system in a structured way. The next sub-sections discuss the backbone of the SSFD framework; the SSFD function model and the SSFD heuristics.

6.2.1 The representation of a function using the SSFD function model

The SSFD function model has been proposed after a critical review of the current state in literature. There are various similarities and differences between the SSFD function model and the existing function modelling approaches.

There are two main differences between the SSFD function model and the established function modelling approaches the FB (see Section 2.3.3.4), the IDEF0 (see Section 2.3.4.1), the C&C^2-A (see Section 2.3.4.4):

- The focus of the FB, the IDEF0, the C&C^2-A is the flow of object per se. For example, in the FB, one of the inputs of the bread toaster functional model is described as "bread" without specifying its characteristics in

Figure 2.11, while the bread toaster SSFD in Figure 4.18 shows the bread as an input, but by measurable attributes which support the articulation and the representation of a function in a coherent way by monitoring the bread attributes. For example, the function "toast bread" in Figure 4.4 shows that the function addresses all specified attributes of the bread including the global attributes "location" and "time", while the FB (as well as the IDEF0 and the C&C²-A) do not take into account the way of achievement of the function.

- The methodologies of the FB, the IDEF0 and the C&C²-A are based on the representation of all flows through the system, including design solutions on a black box which provides a basis for the development of the functional model. While the SSFD function model conceptually includes the design element, only the flow of state transitions through the system are represented to divorce the consideration of function from the consideration of the design solution.

There are several similarities and differences between the SSFD function model and the SBF (see 2.3.5.3), the Statecharts (see Section 2.3.4.3) and object state-transition model of the IDEF3 (see Section 2.3.4.1.2):

- The SSFD function model is coherent with the SBF and the Statecharts as well as object state-transition model of the IDEF3, that is, the notion of attribute is emphasized in the representation of a state.

- While the SBF uses the term "parameter" in the description of a state, the SSFD function model uses the term "attribute" (instead of "parameter") and furthermore differentiates between attributes related to the characteristics of an object (e.g. size) and the global attributes (i.e. time and location). This facilitates the articulation and the representation of both conversion and transmission operations on a state transition.

- The Statecharts and the IDEF3 focus on single parameter of objects. The main difference of the SSFD function model to the SBF is the introduction of a formal way of articulating a function and representing a state, i.e. the function is articulated with related to the object attributes, while the state is represented in a box which has two sections to differentiate between the object and its attributes.

A point of reflection in relation to the use of the SSFD function model for the description of a state is that some terms cannot be immediately described as an object and this could affect the solution-neutral characteristic of the approach. As discussed in Section 3.1.2.1, the proposed SSFD function model exploits the concept of Dori (2002) for the definition of an object, which has been recently adopted as an international standard for automation systems and integration (International Organization for Standardization, 2015). The concept of Dori (2002) enables to describe many things as an object, provided that it possesses measurable attributes. For example, electrical energy in the spoiler SSFD in Figure 5.27 is described both as an attribute of an object (i.e. vehicle) and as an object with measurable attributes. However, mechanical energy in the same SSFD cannot be shown as an object, since it does not possess any attribute, it is an attribute per se. Therefore, it is shown as an attribute of an abstract term (i.e. linkage mechanism) in Figure 5.27, which implies a design solution. While "mechanical energy" is the only case in this thesis, there are various terms with the same issue, e.g. force, heat and kinetic energy.

It is therefore recognized that in order to support a broader practical implementation of the SSFD function model in industrial practice, further research is needed to develop a SSFD ontology. This should also focus on non-measurable attributes which may be related to a functional requirement, e.g. smell of a perfume or the aesthetic appeal of a vehicle.

6.2.2 The SSFD heuristics in function modelling

The SSFD heuristics were proposed as a formal approach for the deployment of the SSFD function model to develop function modelling of a system. The heuristics were mainly adapted from Pahl et. al (2007) and Otto and Wood (2001).

6.2.2.1 Main flow heuristic

For Pahl and Beitz (Pahl et. al, 2007), the "main flow" is the "prevailing flow" through the system, while the dominant flow heuristics of Otto and Wood (2001) focuses on the identification of "dominant" flow by identifying modules on an existing function model. The main flow heuristic of the SSFD boosts the concepts of Pahl et. al (2007) and Otto and Wood (2001) by describing how to identify the flow that is related to the utility of the system to the customer. The practical importance of determining the main flow heuristic is that it supports the hierarchical cascade of functional requirements from customer to sub-functions (i.e. connecting flows and branching flows).

6.2.2.2 Connecting flow heuristic

Pahl and Beitz articulate the connecting flow as “auxiliary flow” which are described with their functions based on design elements of the functions on the main flow, while Otto and Wood (2001) use the term “branching flow heuristic” to describe flows that branch into or out of parallel function chains on a developed function model. The main features of the SSFD connecting flow heuristic can be described as follows:

1) It promotes solution-neutral function modelling by determining a connecting flow as an enabling flow that is required by a conversion operation on the main flow,
2) It differentiates between two types of branching flows by using the term “connecting flow” (representing a flow that branches into the main flow) and “branching flow” (representing a flow that branches out of the main flow).

6.2.2.3 Branching flow heuristic

Pahl and Beitz and Functional Basis of Otto and Wood (2001) do not explicitly address the determination of branching flows of a system. Otto and Wood (2001) describe unintended by-products of a system on a black box. The branching flow heuristic of the SSFD promotes the identification of unintended by-products by focusing on modified object attributes while the system is in use, e.g. the flow of moisture through the bread toaster during the process of toasting a bread can be described by this heuristic (see Figure 4.18), while the functional basis tends to miss this flow (see Figure 2.11).

It is noteworthy that the SSFD heuristics can also be described as a functional reasoning scheme. The heuristics carry out the reasoning process by monitoring the object attributes through the system, which is a key advantage of the SSFD heuristics compared to the other function modelling approaches in literature.

The application of the SSFD heuristics on the case studies “glue gun”, “radiant heater” and “fuel gauge” as well as “bread toaster” highlighted two main points of the SSFD heuristics:

1) The application of the branching flow heuristic on the case studies have revealed that the flow of information cannot be represented as a branching flow due to the fact that it is cannot be described as an unintended by-product. However, once the design elements on the SSFD are known, additional flows (both connecting and branching) on the SSFD can be captured through the analysis of the relevant design element in respect of the customer requirements.

2) The application of the connecting flow heuristic in the case studies has shown that a conversion function on the main flow require a supporting flow depending on its design solution which is conceptually included on the SSFD. For example, the function “convert electrical energy to thermal energy” on the main flow of the radiant heater SSFD in Figure 4.36 does not require a connecting flow due to the way of achievement of the function, i.e. the flow of electrical energy through a design element (i.e. conductor) generates heat. However, the function “toast bread” in the bread toaster SSFD (see Figure 4.18) requires a connecting flow to support the conversion of the bread into the toast by applying heat to bread.

6.2.3 The representation of expected and structure behaviour of a system

Most of the function modelling approaches in literature do not differentiate between expected and structure behaviour (see Section 2.3.5.1) of a system in function modelling, e.g. Task- (Section 2.3.2) and Flow-oriented (Section 2.3.3) approaches. The SSFD is capable of determining and representing both expected behaviour and structure behaviour of a system in a single diagram. This thesis focused on the representation of expected behaviour by developing a SSFD without referring to the design elements. Once the design thesis is complete, the structure behaviour of the system using the SSFD heuristics can be represented by capturing additional flows related to the way of operation of the design elements.

For example, for the bread toaster SSFD in Figure 4.18, locations of the states can be specified if the structural architecture of the toaster is known, as shown in Figure 6.2.

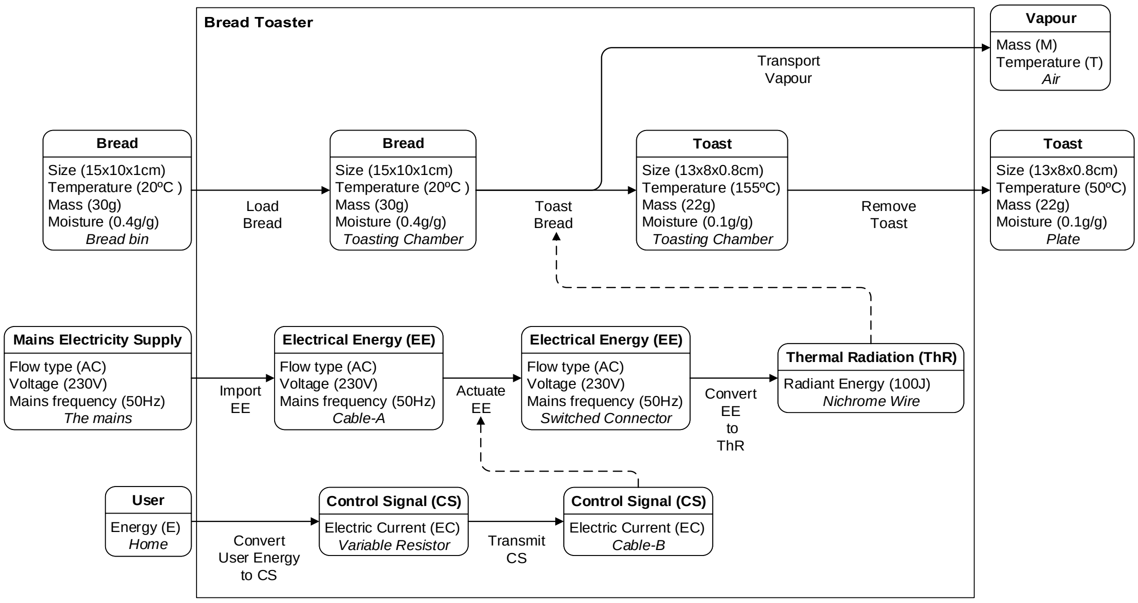


Figure 6.2: An updated version of the bread toaster SSFD

Figure 6.2 shows that the function "load bread" addresses the flow of bread from the bread bin to the toasting chamber where the bread is retained. Coherent with the structure of a household bread toaster, it is assumed that the function "load bread" is achieved by so-called "grates mechanism" which centres and lowers the bread in the toasting chamber. From a device centric perspective, the SSFD for the function "load bread" can be developed using the SSFD heuristics, as illustrated in Figure 6.3.

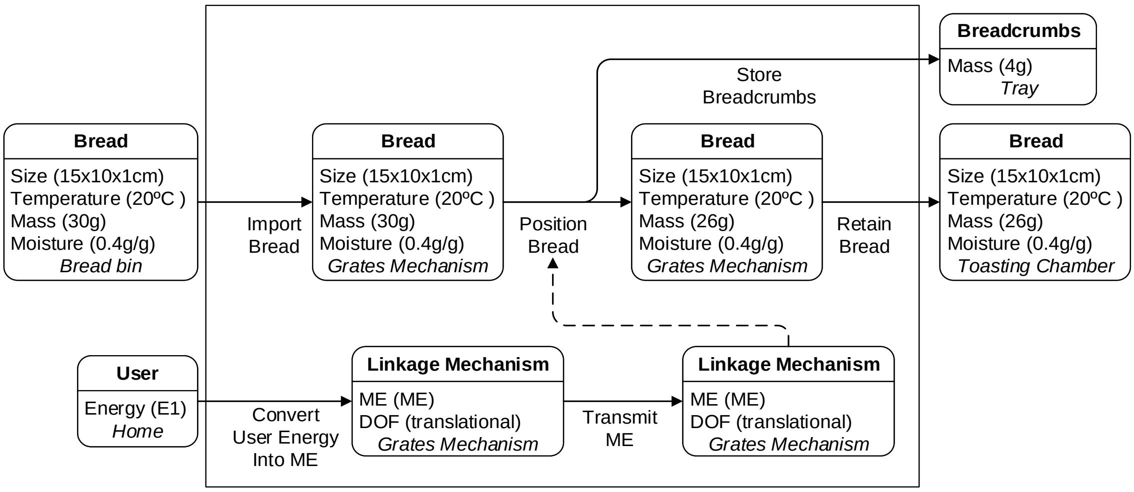


Figure 6.3: SSFD for the function "load bread"

The flow of the bread from the bin to the chamber in Figure 6.3 is summarized as follows:

- Similar to the function "load bread" in Figure 6.2, the flow starts by bringing in the bread from outside the system boundary (i.e. import bread).

- This follows the vertical and the horizontal alignment of the bread within the toasting chamber (i.e. position bread).

- The bread is kept inside the toasting chamber during the process of toasting (i.e. retain bread).

- The user energy is converted into mechanical energy at a linkage mechanism. The mechanical energy is transmitted to the grates mechanism for the achievement of the function "position bread", as shown in Figure 6.3.

- Breadcrumbs are to become detached from the bread (i.e. store breadcrumbs) due to mechanical contact between the bread and the grates mechanism as a result of the function "position bread".

The SSFD in Figure 6.3 can be integrated into the bread toaster SSFD in Figure 6.2 in respect of the function "load bread". Figure 6.4 represent an updated bread toaster SSFD. The user energy values in Figure 6.4 are articulated as "E1" and "E2".

Figure 6.4 shows both expected behaviour and structure behaviour of the bread toaster in relation to the function “load bread”. The “laws of conservation” for the functions “store breadcrumbs” (1) and “position bread” (2) can be represented by the following equations:

Breadcrumbs (4g, Tray) = f(Bread (15x10x1cm,20ºC,30g,0.4g/g,Grates Mechanism) - Bread(15x10x1cm,20ºC,26g,0.4g/g,Grates Mechanism)) (1)

Bread(15x10x1cm,20ºC,26g,0.4g/g,Grates Mechanism) = f(Bread (15x10x1cm,20ºC,30g,0.4g/g,Grates Mechanism). Linkage Mechanism (ME3, Translational, Grates Mechanism)) (2)

The flow of information as a branching flow can be represented by following the same methodology. However, this needs to be demonstrated with further case studies on the basis of system requirement lists - beyond the scope of this research.

6.3 Development of SSFD for complex multidisciplinary systems with multiple operation modes

6.3.1 The SSFD in the analysis of complex multidisciplinary systems

As discussed in Section 3.3, complexity and multidisciplinarity of a system can be associated with the flows of material, energy and information through the system. The system gets complicated as it addresses more the flows of material, energy and information. A system regarding a particular engineering discipline includes particular flows, e.g. the flow of energy for an electrical system; therefore, a multidisciplinary system can include multiple flows of material, energy and information. It can thus be suggested that there is a direct correlation between the complexity and the multidisciplinarity.

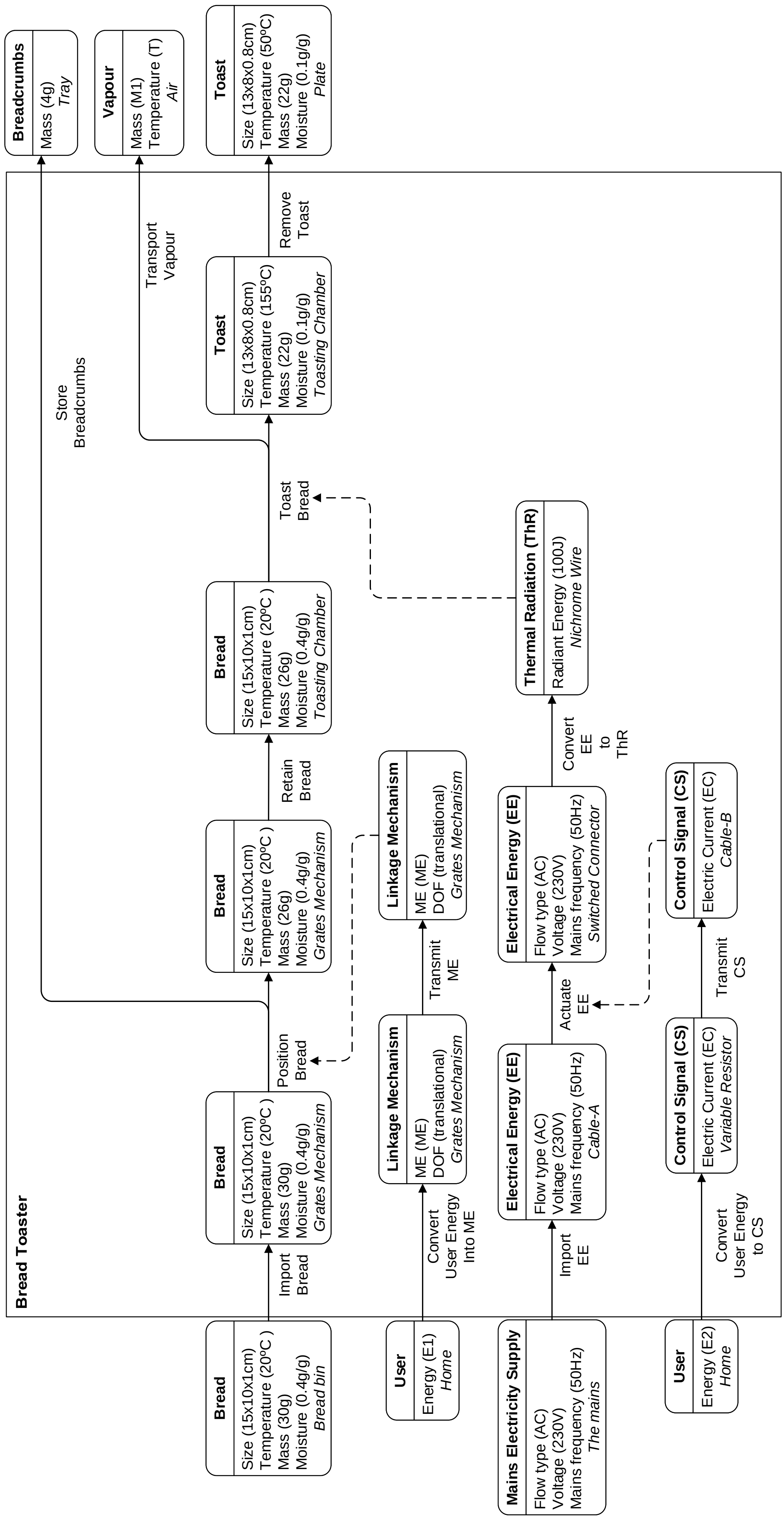


Figure 6.4: An updated version of the bread toaster SSFD

The FB and the IDEF0 are well-known approaches in function modelling considering they run through many published work. These approaches define all multi-disciplinary features of a system on a black box and develop functional model of the system on the basis of the black box. For example, the FB model of the bread toaster in Figure 2.11 requires the development of function chains for the input flows of “electrical energy”, “hand”, “darkness control”, “bread”, “human energy” and “weight”. One of more function chains can be related to different features of the toaster such as the chains of “bread”, “hand” and “human energy” for mechanical features, while the chain of “darkness control” is about control mechanism of the toaster. All these function chains are aggregated into a single model, however, as discussed in Section 2.5.2, it is not clear how to connect distinct function chains together in the FB. Similarly, the top-level context diagram of the IDEF0 can include the same input flows through its bounding arrows “input” and “control” as well as “mechanism”. These inputs flows are allocated to child diagrams. As mentioned in Section 2.5.3, the main problem of the IDEF0 is lack of a coherent guideline on the use of the approach, namely that the approach does not specify a starting point for the development of the child diagrams which represent sub-functions.

As discussed in Section 6.2, the SSFD framework develops functional model of a system through functional reasoning. In terms of the analysis of the bread toaster in Chapter 4.1, the framework determines multi-disciplinary features of the toaster on the basis of the analysis of the main flow (i.e. bread) using the SSFD heuristics. Figure 6.4 shows mechanical, electrical and control features of the toaster with reference to the flow of material, energy and information. For example, the flow of bread through the toaster and the conversion of user energy into mechanical energy in respect of the function “position bread” on the main flow are related to mechanical feature of the toaster. All features are determined by mapping object attributes through the SSFD heuristics in a solution-neutral way. ARS and EVP SSFDs in Section 5.4 represent multi-disciplinary characteristics of these systems in respect of multiple operation modes. For example, ARS SSFD in Figure 5.27 shows different operation modes of mechanical, electrical and control features of the ARS by representing the flow of spoiler, electrical energy, driver (input) and control signal with different attribute values with respect to relevant ARS operation modes.

6.3.2 Modelling multiple operation modes in the SSFD using the ESD and the SSFD fork node

Liu et al. (2015) point out that modern mechatronic products including electric home appliances and automobiles incorporate multiple modes. The same issue has been discussed in Chapter 5 in terms of complexity of systems. Complex systems can have multiple modes of operation, which each have different functional requirements. For example, the analysis of the electric vehicle powertrain in Section 5.4.2 showed that the use cases “charge EV”, “move EV” and “power EV accessories” have different functional requirements, which are related to each other. Most functional representation models do not adequately capture this in a concise functional model of the system, they mostly focus on the development of the functional model on the basis of the analysis of the overall functional requirement which is commonly represented in a black box. Figure 6.5 summarizes the way of developing a functional model for four prominent function modelling approaches in literature.

The first steps of the FB, the IDEF0 and the method of Pahl and Beitz in Figure 6.5 show that these approaches focus the development of functional model of systems based on one mode of operation.

FB (Stone and Wood, 2000)	**IDEF0 (Buede, 2009)**
1) Generate Black Box model 2) Create function chains for each input flow 2.1) Express sub-functions in a common functional basis 2.2) Order function chains with respect to time 3) Aggregate function chains into a functional model	1) Represent the top-level context diagram 2) Develop child diagrams on the basis of the top-level context diagram
C&C²-A (Matthiesen and Ruckpaul, 2012)	**Pahl and Beitz (Pahl et. al, 2007)**
1) Define the relevant parts of the system and its borders 2) Detect the relevant WSPs which are mandatory for fulfilling the main function 3) Identify and cluster the CSSs which contribute to the building of a WSP 4) Identify several logical states of the CSS-cluster 5) Arrange a logical structure for representing all correlations and dependencies between the CSS-clusters	1) Define the overall function on a block diagram (i.e. black box) 2) Break the overall function down into sub-functions 2.1) Determine the main functions 2.2) Determine the auxiliary functions

Figure 6.5: Function modelling steps of four prominent function modelling approaches

The C&C²-A promotes the development of existing systems (see step 1 in Figure 6.5). The approach supports the development of systems with multiple operation modes (see Step 4 in Figure 6.5) in respect of the main function (see Step 2 in Figure 6.5). However, a graphical format of the complete functional representation of a system with multiple operation modes is not available.

On the other hand, systems engineering tools, in particular sequence diagrams (see Section 2.3.6.5.2 and Section 5.2.2) represent different modes of operations of a system in terms of use cases and identify functional requirements for each use case by focusing on the sequence of activities. However, a compact model of functional representation of multiple use cases of a system is not available in a graphical format.

This research adapted the features of current sequence diagrams to provide a basis for the development of SSFDs of multiple operation modes of the same system by taking into account the sequences of states and functions. The need for the development of a new sequence diagram tool, herein named Enhanced Sequence Diagram (ESD), was based on the shortcomings of the conventional sequence diagrams in literature. The SysML sequence diagram focuses on message-based interactions, while the attempts of Zingel et al. (2012) and Piques (2014) do not reflect multiple flows through the system. At high level, the SysML sequence diagram and the ESD possess similar features, i.e. the representation of an actor in a rectangle with a line descending from the base of the actor and the representation of the flow of interactions on this line vertically in respect of time. The necessity of extending the SysML sequence diagram with the following three notions was based on the need to represent a SSFD on the basis of a sequence diagram. Key features of the proposed framework include:

1) Coherent with the SSFD function model, the ESD describes the actor with measurable attributes and articulates a function in relation to the interaction between actors. This is an important enhancement over the conventional sequence diagrams as it facilitates the determination of a function on a structured basis by referring to the actor attributes.

2) Coherent with the SSFD function model, within the ESD the system under consideration is considered as consisting of multiple flowlines of material, energy and information. The flows through the system under consideration are then represented by mapping the actor attributes, and links between different flowlines are done consistent with the SSFD heuristics. This enhances considerably the conventional sequence diagrams by addressing multiple flows (and functional requirements) through the system in a single diagram.

3) The notion of "scope lines" was introduced to define the boundaries of the diagram. The main feature of this notion is to take into account the links between developed SSFDs for multiple operations of the same system, e.g. the output of one SSFD may be the input of the other SSFD.

"Time" is represented in the SysML sequence diagram in terms of the notions of "duration" and "observation". The ESD represents time implicitly by mapping the flows vertically - similar to the conventional "lifeline" representation of the system/actor in the SysML sequence diagram (Friedenthal et al., 2008). It is important to integrate this feature into the SSFD. As it would strengthen the SSFD significantly, since it is difficult to represent time in a SSFD by only relying on the SSFD heuristics due to the fact that the heuristics do not take into account the sequences of states and functions with respect to time, as mentioned before.

The implementation of the SSFD heuristics on the case studies "bread toaster", "glue gun", "radiant heater" and "fuel gauge" in Chapter 4 were based on one mode of operation. The necessity of aggregating SSFDs for multiple operation modes of the same system into a single diagram required the introduction of the SSFD fork node by adopting the strengths of the current nodes. The ESD was illustrated on an industrial case study of a front view split camera and implemented along with the SSFD fork node on the electric vehicle powertrain and the active rear spoiler for the representation of function modelling of systems with multiple operation modes. The application to the case studies including the electric vehicle powertrain and the active rear spoiler have provided valuable insights into function modelling of complex multidisciplinary systems using the SSFD framework.

In terms of the key weaknesses of the current function modelling approaches highlighted in Section 2.5, the contribution of the SSFD framework can be summarized as follows:

1) The SSFD heuristics provided a formal approach of combining different chains of functions into a single diagram.

2) One of the novel features of the SSFD is to identify possible unintended by-products while the system is in use through the branching flow heuristics. However, as discussed before, the applicability of the branching flow heuristics on the case studies of this thesis were limited to the flows of material and energy. This is an important issue for future research.

3) The introduction of the ESD and the SSFD fork node have extended to use of the SSFD heuristics to develop functional model of systems with multiple operation modes.

4) The SSFD uses adopted nodes of the existent approaches. Unlike the SysML state machine diagram and the OPM, the development of the SSFD does not require a large number of nodes. To the extent of my experience, the diagram is easy to use. However, more research on this issue needs to be undertaken in terms of the test of the take-up of the approach in industry. Because of the fact that the SSFD can be represented using the nodes of the SysML diagrams, the practical applicability of the SSFD can be promoted by exploiting SysML.

6.4 The use of the SSFD in systems engineering design

6.4.1 The failure mode avoidance process

The proposed SSFD methodology integrates well with the Failure Mode Avoidance (FMA) framework. Figure 6.6 illustrates the FMA process.

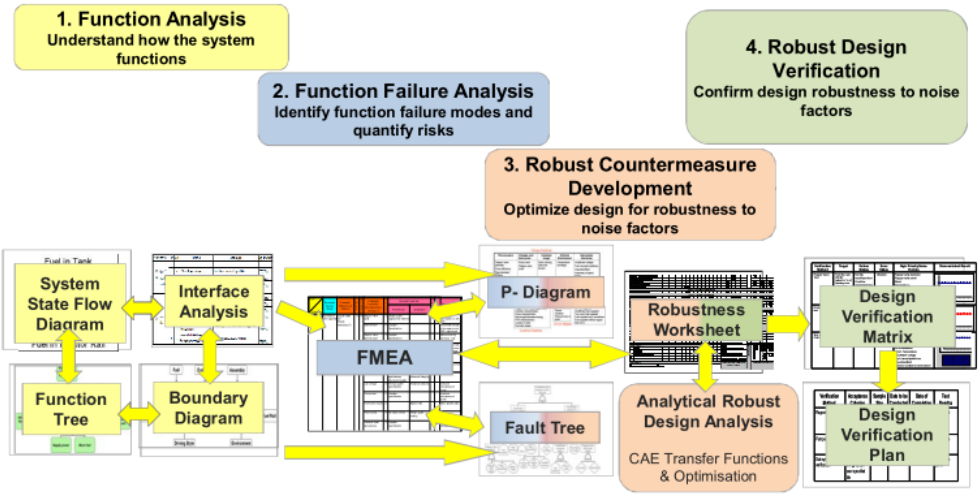


Figure 6.6: FMA process (Campean and Henshall, 2012a, p.47)

The SSFD plays an important role in the vertical and the horizontal deployment of the FMA process. Coherent with the Systems Engineering "V", the developed SSFD framework supports the vertical deployment of the FMA process by underpinning hierarchical decomposition of functions with a strong focus on customer required functionality. The framework provides a rigorous definition on how the SSFD can be developed and deployed across systems levels. Section 6.2.3 exemplified this on the basis of the analysis of the bread toaster. The horizontal deployment of the FMA process starts by the development of the SSFD and ends with the documentation of a design verification plan, as shown in Figure 6.6. The SSFD framework supports the horizontal deployment of the FMA process by underpinning the development of other tools in the function analysis step of the FMA in Figure 6.6, i.e. Function Tree, Boundary Diagram and Interface Analysis.

In terms of the development of a function tree, a function tree for a system with multiple modes can be developed through the SSFD framework. By doing so, the function tree provides a detailed and structured breakdown of system functional requirements in respect of each operation mode. This can be observed by comparing the EVP function tree in Figure 5.19 and the EVP function tree in Campean et al. (2011) (see Appendix E). Furthermore, the function tree in Figure 5.19 differentiates between the main flow sub-functions and the connecting flow sub-functions of the EVP functional requirements (i.e. use cases).

A system boundary diagram shows the design elements which contribute directly to achieving the functions on the SSFD in terms of the flows of energy, material and information (Henshall and Campean, 2009). A boundary diagram for a system with multiple modes can be represented using the SSFD framework. A system boundary diagram for the EVP is represented in Figure 6.7.

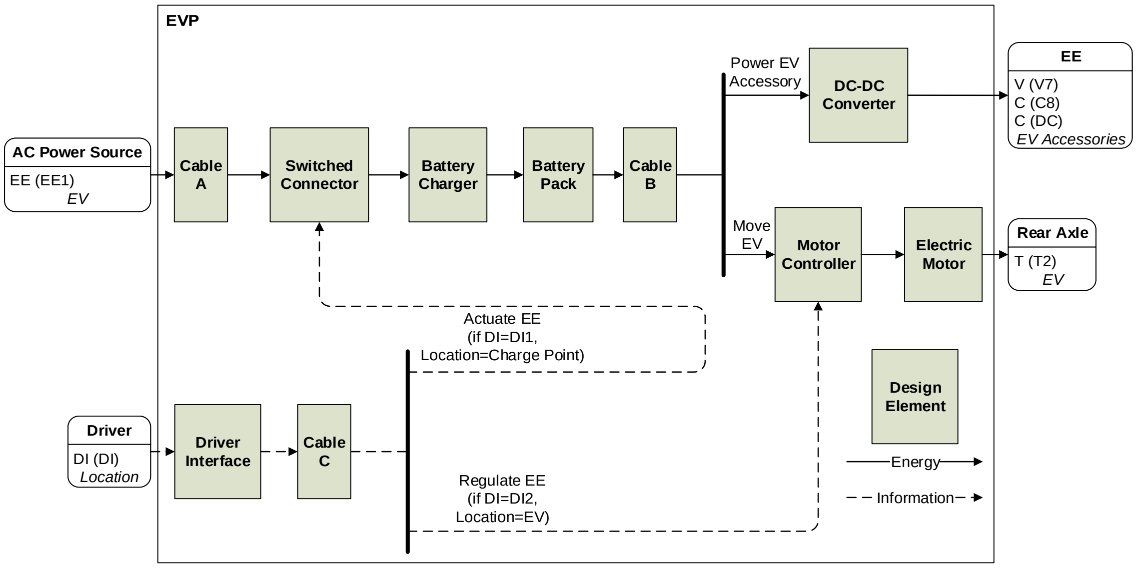


Figure 6.7: Boundary Diagram for the EVP

The functionality of the fork node in Figure 6.7 is the same as with the SSFD fork node, that is, it enables to represent multiple operation modes of the EVP. Similar to the EVP SSFD in Appendix F, the EVP boundary diagram in Campean et al. (2011) (see Appendix G) represents the achievements of the EVP functional requirements concurrently, e.g. it shows that the driver can drive the vehicle, while the vehicle is on charge, which is practically not possible.

The choice of design elements in Figure 6.7 is based on the structure of the EVP, which is detailed in Campean et al. (2011). According to the SSFD function model, a design element can address one function, however it may be a case that a design element can achieve multiple functions, e.g. DC-DC converter fulfils the functions “Convert HV/DC into LV/DC” and “Transmit EE” on the EVP SSFD in Figure 5.18. Therefore, it would be useful to carry out further work for the determination of the design elements on the SSFD, i.e. design synthesis.

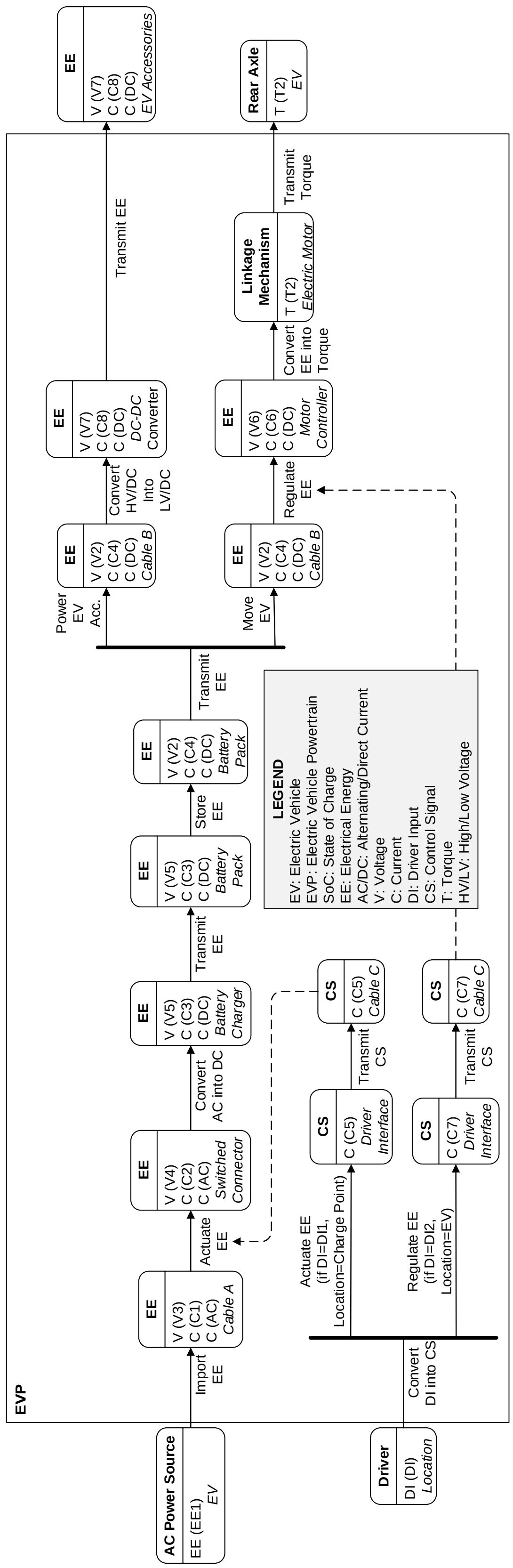


Figure 6.8: An updated version of the EVP SSFD

Location attributes of the states on the EVP SSFD in Figure 5.18 can be updated based on the design elements on the EVP boundary diagram in Figure 6.7. Figure 6.8 presents this updated version of the SSFD, with the structural design elements explicitly shown as “location”. The states in Figure 6.8 identify key parameters of the design elements, e.g. Voltage (V3) and Current (C1) of “Cable A” in related to the function “import EE”.

While the FMA process in Figure 6.6 does not show a direct link between the SSFD and the interface analysis (interface matrix/table) of the function analysis step, it can be suggested that there is a mutual link between them.

- The SSFD promotes the description of interface requirements and the identification of interface functions to address these requirements in the interface analysis table.

- The interface analysis table can support the identification of intermediate state transitions (and therefore functions) in the SSFD (see Section 6.4.2).

Figure 6.9 shows an interface analysis table for two internal interfaces of the EVP in respect of the high level function (i.e. use case) “charge EV”: Battery Charger and Battery Pack. The table includes the following elements (from left to right):

- Interface name,

- A description of the interaction,

- The type of the interaction, where Energy and Information are denoted by “E” and “I” in Figure 6.9 respectively,

- A statement of the engineering function required to manage the interaction,

- Destinations of the interaction,

- The target attribute value of the interaction,

- An evaluation of the effect of the interaction on the high level function, where “+2” in Figure 6.9 denotes that the exchange must be provided to support the high level function,

- Related high level function is also documented on the table.

Interface	Generic Interaction Scenario Description	Interaction Exchange Type	Interaction Functions and Exchanges			From	To	Requirement Specification (Target Attribute(Value))	Interaction Criticality	High Level Function
			Verb	Object						
				Input	Output					
Battery Pack & Battery Charger	The flow of Electrical Energy from Battery Charger to Battery Pack	E	Transmit		Electrical Energy	Battery Charger	Battery Pack	Voltage (V5) Current (C3) Current (DC)	2	Charge EV
	Battery Pack Temperature Info from Battery Pack to Battery Charger	I	Measure	Battery Pack SoC		Battery Pack		Current (C9)	2	
		I	Transmit		Battery Pack SoC	Battery Pack	Battery Charger	Current (C9)	2	

Figure 6.9: Example of interface analysis table for the EVP (adapted from Yildirim and Campean, 2013 (content) and Uddin et al., 2015 (template))

The effect of the interface description "the flow of Electrical Energy from Battery Charger to Battery Pack" is shown as "2" in Figure 6.9, which demonstrates that the interaction is vital for the use case "charge EV". This exchange corresponds to a state transition on the main flow of the EVP SSFD in Figure 6.8 and the function "transmit EE" in the same figure addresses the state transition. The flow of information from the SSFD in Figure 6.8 to the interface analysis table in Figure 6.9 is shown in Figure 6.10 for the function "Transmit Electrical Energy" based on excerpts from the SSFD and the interface analysis table.

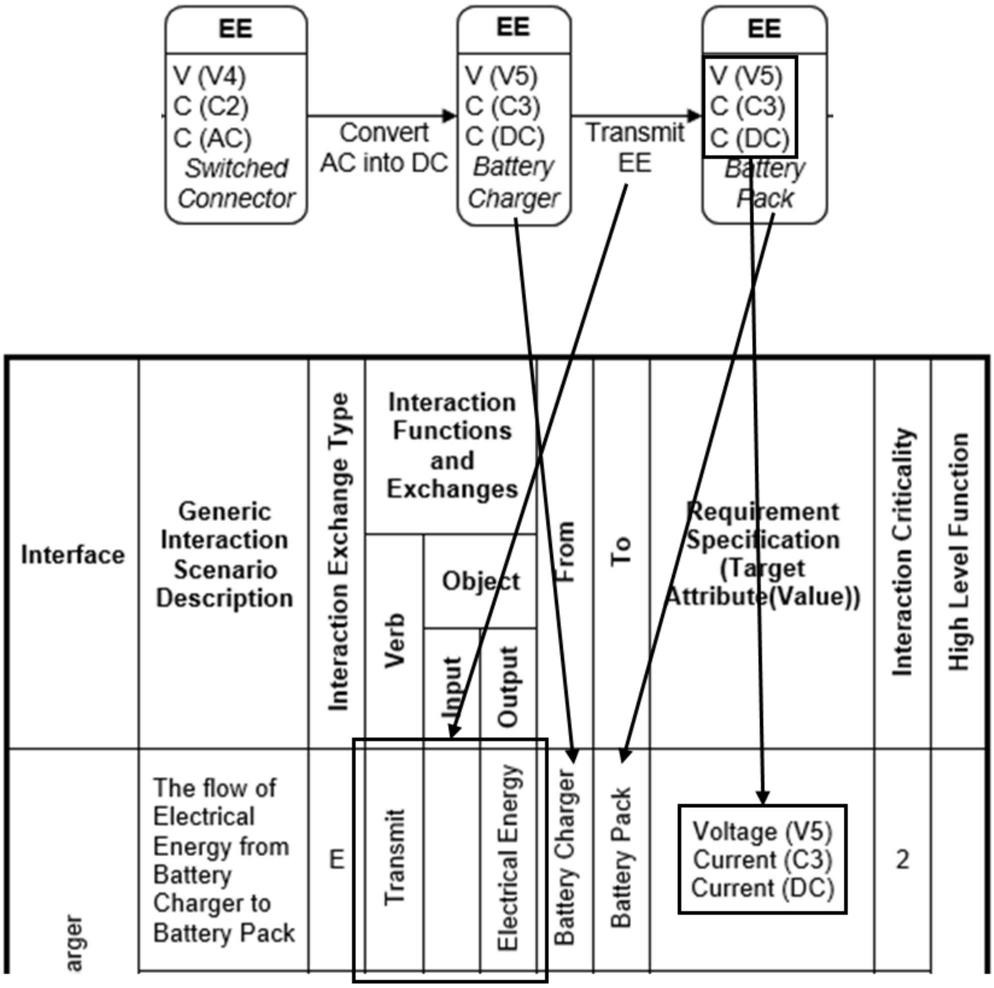


Figure 6.10: The flow information from the SSFD to the interface analysis table for the function “transmit EE”

Figure 6.10 shows that the function “transmit EE” in the SSFD corresponds to an interface function in the interface analysis table in terms of a verb and an object, while the global attribute “location” in the SSFD is represented as the destination of the interface in the interface analysis table. The interface analysis table shows the output object attributes of the function “transmit EE” in the SSFD as “requirement specification (target attribute value)”.

6.4.2 Systems engineering design integration and engineering change management

The research work on the use of an enhanced interface analysis method for engineering change management (Yildirim and Campean, 2013) points out that the flow of information from the interface analysis table to the SSFD can support the development of the SSFD in terms of systems engineering design integration and engineering change management.

Regarding systems engineering design integration, the interface functions “Measure Battery Pack State of Charge (SoC)” and “Transmit Battery Pack SoC Info to Battery Charger” in Figure 6.9 cannot directly be identified through the SSFD heuristics, however the EVP SSFD in Figure 6.8 can be refined for the inclusion of these interface functions through the SSFD heuristics. This requires the analysis of the complete interface analysis table of the EVP. By doing so, the refined SSFD can represent structural behaviour of the “Battery Charger” and the “Battery Pack”.

In terms of engineering change management, the change of a design element in a system can affect structural behaviour and expected behaviour of the system which can be represented by the SSFD. The impact of the engineering change on the SSFD must be evaluated through the interface analysis table in terms of the internal complexity (i.e. within the design element) and transmitted complexity to the interfacing subsystems / components. For example, in the case of the change of the EVP battery pack, an internal requirement cascaded to the Battery Pack is to “Transmit Battery Pack SoC Info to Battery Charger” in Figure 6.9, which is presumably achieved by a temperature sensing system. Thus, it is required to evaluate;

1) Internal complexity - i.e. whether the temperature sensing system can be carried over or a new system is required following the change of the battery capacity. Both situations may require the modification of the relevant flow (i.e. information flow related to the temperature sensing system) in the EVP SSFD,
2) Transmitted complexity - whether there is a requirement to have an engineering design change for the battery charger in order to fulfill the interface function “Transmit Battery Pack SoC Info to Battery Charger” with the new battery pack. Like the first point, this could lead to the refinement of the EVP SSFD on the basis of the determined interface functions in the EVP interface analysis table.

By following the same practice in Figure 6.10 in reverse, the flow of information from the interface analysis table to the SSFD can support the development of the SSFD in terms of representing additional flows as support flows through the SSFD heuristics. This ensures that relationships between sub-functions can clearly be documented in the SSFD. However, further studies are needed to develop a full picture of mapping of functions between the SSFD and the interface analysis table.

7. Conclusions and Further Work

7.1 Review of Research Contributions

The original contributions of the research presented in this thesis can be summarized as follows:

- The thesis makes a meaningful contribution to research by introducing a strong theoretical underpinning for the SSFD framework, including definitions of key concepts and elements (“state” (i.e. object, attribute and value), “function” and function representation model based on the “triad” - as illustrated in Figure 3.10) underpinning the framework. A more coherent and comprehensive graphical representation of the SSFD has been provided by including local attributes describing the state of the object as well as global attributes of time and location (Chapter 3).

- A coherent functional reasoning scheme has been introduced based on the SSFD heuristics to guide the development of function model of a system in a structured yet practical manner (Chapter 4).

- The main flow heuristic was introduced to support the cascade of functional requirements from customer to sub-functions by focusing on customer required functionality, while the branching flow heuristic makes a general contribution to research by promoting the identification of unintended by-products while the system is in use (Chapter 4).

- An “Enhanced Sequence Diagram” tool and methodology has been introduced, which is a significant extension of the current sequence diagram method, enabling the description and mapping of individual flows of material, energy and information associated with the timeline of an actor. This provides a much richer (accurate and detailed) mapping of sequences of events in describing the functions of a system (Chapter 5).

- A valuable contribution was also made by introducing a methodology for the development of the SSFD in conjunction with the “Enhanced Sequence Diagram”. This is an important development that paves the way for integration of the SSFD with the SysML systems engineering diagrams, but also enhances the ability of the SSFD to accurately represent conditional flows (Chapter 5).

- An enhanced SSFD representation and methodology was presented to support the representation of multi-mode system functionality in a compact SSFD model of the overall system function. The key enhancing elements for the SSFD framework are the fork node to support representation of conditional flows, and the parametric mapping of the SSFD to support the function traceability across a complex system (Chapter 5).

- The thesis provides a sound case for the theoretical validity of the SSFD framework by developing its key concepts and elements based on critical analysis of other established function modelling frameworks. Empirical validity of the framework is provided through desktop and real world engineering case studies (Chapter 2-to-6).

- An analysis has been provided on how successive levels of analysis can be integrated within a SSFD to offer consistent integration of multiple levels of analysis through a nested system structure (Chapter 6).

- The thesis has also provided an analysis on how the expected behaviour and the structure behaviour of a system can be represented within a SSFD (Chapter 6).

7.2 Conclusions

Based on the research results, analysis and discussion presented in this thesis the following conclusions can be drawn:

- The SSFD function model presented in this thesis forms the basis of the SSFD framework by providing a rigorous definition and representation of the key elements of “state” and “function” based on the proposed function representation concept “triad”. The model is validated throughout desktop and real world engineering case studies.

- The SSFD heuristics provides a structured guideline of how function model of a system can be developed using the SSFD function model on a practical basis. The applicability of the SSFD heuristics to the flows of material, energy and information was validated on desktop case studies.

- The SSFD framework presented in this thesis provides a rigorous and coherent function modelling framework for the analysis of complex multidisciplinary systems with multiple operation modes by supporting the representation multiple flows of different system operation modes in a single diagram. The use of the framework in function modelling of multiple operation modes was validated on real world engineering case studies.

- This thesis has shown that the SSFD framework is capable of representing the expected behaviour and the structure behaviour of a system in a single diagram. While the research focused on the representation of the expected behaviour in thesis, it has been demonstrated that the SSFD also provides a framework for the representation of the structure behaviour of a system along with the expected behaviour in the same diagram.

- The presented analysis in the thesis has demonstrated that the SSFD framework supports consistent integration of multiple levels of analysis through a nested system structure.

- The introduced methodology for the development of the SSFD in conjunction with the “Enhanced Sequence Diagram” and “Use Case Diagram” promotes the integration of the SSFD with the SysML diagrams.

7.3 Further Work

Research opportunities arose from this thesis can be summarized as follows:

- Further research is needed to develop an SSFD ontology with a strong focus on non-measurable attributes (e.g. the aesthetic appeal of a vehicle body) in order to support a broader practical implementation of the SSFD function model in industrial practice.

- The representation of the flow of information as a branching flow in related to the structure behaviour of a system on the SSFD requires further work.
- In further research, “time” could be shown on the SSFD through the ESD.
- It is required to test the take-up of the developed approach in industry. The BEQIC FMA process has been extensively taught and deployed within an industrial environment. There will be plenty of opportunity to test the SSFD framework in the context of the FMA process. This will enable both the test of the framework per se and its compatibility with the FMA process.
- Considering the SSFD function model and the structure of the SSFD of a complex system, a structured approach should be established to determine design elements on the SSFD, i.e. design synthesis.
- Further research should be undertaken to determine and represent additional flows in the SSFD through the interface analysis table.

References

Ahmed, A. and Dori, D. (2009) Conceptual modeling of complex systems via object process methodology. *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, 2(1). pp. 171-176.

Albers, A., Burkardt, N. and Ohmer, M. (2004) Principles for design on the abstract level of the contact & channel model. *Proceedings of the TMCE.* Lausanne, 13-17 April. Rotterdam: Millpress, pp. 87-94.

Albers, A., Burkardt, N. and Ohmer, M. (2005) Approaches for the synthesis of technical systems within the contact and channel model. *International conference on engineering design – ICED*. Melbourne.15-18 August.

Albers, A., Alink, T., Matthiesen, S. and Thau, S. (2008a) Support of system analyses and improvement in industrial design through the contact&channel model. *International design conference – DESIGN.* Dubrovnik, 19-22 May.

Albers, A., Deigendesch, T. and Alink, T. (2008b) Support of design engineering activity - the contact and channel model in the context of problem solving and the role of modelling. *Proceedings of the International Design Conference.* Dubrovnik, 19-22 May, pp.97-102.

Albers, A., Braun, A., Clarkson, P.J., Enkler, H-J and Wynn, D. (2009a) Contact and channel modelling to support early design of technical systems. *International Conference on Engineering Design – ICED.* Stanford, 24-27 August.

Albers, A., Sedchaicharn, K., Sauter, C. and Burger, W. (2009b) A method to define a product architecture early in product development using a contact and channel model. *Proceedings of the International Conference on Engineering Design.* Stanford, 24-27 August, pp.241-252.

Albers, A., Braun, A., Sadowski, E., Wyatt, D.F., Wynn, D.C. and Clarkson, P.J. (2010) Contact and channel modelling using part and function libraries in a function-based design approach. *Proceedings of ASME 2010 international design engineering technical conferences & computers and information in engineering conference – IDETC/CIE*. Quebec, 15-18 August.

Albers, A. and Zingel, C. (2011) Interdisciplinary systems modelling using the contact & channel-model for SysML. *International conference on engineering design.* Technical University of Denmark: 15-18 August.

Albers, A., Ohmer, M. and Alink, T. (2011a) Accessibility of the innovative principles to further levels of abstraction in product development. *Procedia Engineering*, 9, pp.222–235.

Albers, A., Oerding, J. and Alink, T. (2011b) Abstract objectives can become more tangible with the contact and channel model (C&CM). *Models and Methods for Variation Management in Global Product Development: proceedings of the 20th CIRP design conference.* Ecole Centrale de Nantes, Nantes, 19-21 April. Berlin: Springer, pp. 203-213.

Albers, A. and Zingel, C. (2013) Extending SysML for engineering designers by integration of the contact & channel – approach (C&C²-A) for function-based modelling of technical systems. *Procedia Computer Science*, 16, pp. 353-362.

Albers A., Wagner, D., Hammami, W., Spadinger, M. and Hofler, T. (2014) Integration of design methods for the product generation development of electric energy storage systems. *Design Methods: Proceedings of the International Design Conference.* Dubrovnik, 19-22 May, pp. 101-110.

Altshuller, G. (1984) *Creativity as an Exact Science*. The Netherlands: Gordon and Branch Publishers.

Alur, R., Kanade, K., Ramesh, S. and Shashidhar, K.C. (2008) Symbolic analysis for improving simulation coverage of simulink/stateflow models. *Proceedings of the 8th ACM International Conference on Embedded Software.* New York. pp. 89–98.

Alvarez Cabrera, A.A., Erden, M.S. and Tomiyama, T. (2009) On the potential of function-behaviour-state (FBS) methodology for the integration of modelling tools. *Competitive Design: Proceedings of the 19th CIRP Design Conference.* Cranfield, 30-31 March, pp.412-419.

Arlitt, R.M., Balinski, K.D., Dagli, C.H. and Grantham, K. (2011) Functional analysis of systems using a Functional Basis. *Systems Conference (SysCon), 2011 IEEE International.* 4-7 April 2011. DOI: 10.1109/SYSCON.2011.5929052 (Accessed: 05/02/2013).

August, E., Eckert, C. & Clarkson, P.J. (2005) Using design structure matrices in visualising design processes. *Studies in Multidisciplinarity*, 2(22), pp. 305-317.

Bertsche, B. (2008) *Reliability in Automotive and Mechanical Engineering.* Berlin: Springer-Verlag.

Birolini, A. (2010) *Reliability Engineering: Theory and Practice.* 6th Edition. Springer-Verlag.

Blessing, L.T.M. and Chakrabarti, A. (2009) *DRM: A Research Design Methodology*. Springer-Verlag.

Bobrow, D. G. (1984) Qualitative reasoning about physical systems: an introduction. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM)*, 24(1-3), pp.1-5.

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

Brady, T.K. (2002) *Utilization of dependency structure matrix analysis to assess implementation of NASA`s complex technical projects.* MSc thesis. Massachusetts Institute of Technology.

Browning, T. R. (2001) Applying the design structure matrix to system decomposition and integration problems: A review and new directions*. IEEE Transactions on Engineering Management,* 48(3), pp. 292-306.

Bock, C. (2006) SysML and UML 2.0 Support for Activity Modelling. *Journal of International Council of Systems Engineering*, 9(2), pp.160-186.

Buede, D.M. (2009) *The engineering design of systems: models and methods.* 2nd edn. New Jersey: Wiley.

Campean, F., and Henshall, E. (2008) *A function failure approach to fault tree analysis for automotive systems.* Society of Automotive Engineers Technical paper: 2008-01-0846.

Campean, I.F., Henshall, E. and Brunson, D. (2010) Failure mode avoidance paradigm in automotive engineering design. *International Congress on Automotive and Transport Engineering - CONAT, Conference Proceedings.* Brasov, 25th-27th Oct 2010, pp.207-214.

Campean, I.F., Henshall, E., Brunson, D., Day, A., McLellan, R., &Hartley, J. (2011) *A structured approach for function analysis of complex automotive systems.* Society of Automotive Engineers Technical paper: 2011-01-1268.

Campean, I.F. and Henshall, E.J. (2012a) The functional basis for failure mode avoidance in automotive systems engineering design. *2nd International conference on modelling and management of engineering processes (MMEP).*Cambridge, UK, 29-30 November.

Campean, I.F. and Henshall, E.J. (2012b) Systems engineering design through failure mode avoidance - an automotive industry perspective. *Through Life Engineering Services TESConf 2012.* Cranfield, UK, 05-06 November.

Campean, I., Henshall, E. and Rutter, B. (2013a) Systems Engineering Excellence Through Design: An Integrated Approach Based on Failure Mode Avoidance. *SAE Int. J. Mater. Manf.,* 6(3), pp. 389-401.

Campean, F., Henshall, E., Yildirim, U., Uddin, A. and Williams, H. (2013b) A structured approach for function based decomposition of complex multi-disciplinary automotive systems. *Smart product engineering: the 23th CIRP design conference.* Bochum, Germany, 11-13 March.

Chakrabarti, A. and Bligh, T.P. (2001) A Scheme for Functional Reasoning in Conceptual Design. *Design Studies*, 22(6), pp. 493–517.

Chandrasekaran, B. and Josephson, J.R. (2000) Function in device representation. *Engineering with Computers*, 16, pp. 162-177.

Chiang, W-C, Pennathur, A. and Mital, A. (2001) Designing and manufacturing consumer products for functionality: a literature review of current function definitions and design support tools. *Integrated Manufacturing Systems*, 12(6), pp. 430-448.

Chiriac, N., Holtta-Otto, K., Lysy, D. and Suh, ES (2011) Three approaches to complex system decomposition. *13th International Dependency and Structure Modelling Conference (DSM).* 14-15 September, Cambridge, Massachusetts, USA.

Chmarra M.K., Alvarez Cabrera A.A., van Beek T., D'Amelio V., Erden M.S. and Tomiyama T. (2008) Revisiting the divide and conquer strategy to deal with complexity in product design. *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Application*, pp.393-398.

Clarkson, P. J., Simons, C., and Eckert, C. (2001a) Predicting change propagation in complex design. *Proceedings of Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 9-12 September, Pittsburgh.

Clarkson, J., Simons, C., & Eckert, C. (2001b) Change prediction for product redesign. *International Conference on Design-ICED, Conference.* 21- 23 August, Glasgow.

Cook, H.E. and Wissmann, L.A. (2007) *Value driven product planning and systems engineering.* London: Springer-Verlag.

Crilly, N. (2012) Function propagation through nested systems. *Design Studies*, 34(2), pp.216-242.

Cross, N. (2000) *Engineering design methods: strategies for product design.* 3rd edn. Chichester: John Wiley&Sons Ltd.

D`Amelio, V. and Tomiyama, T. (2007) Predicting the unpredictable problems in mechatronics design. *International Conference on Engineering Design.* Paris, 28-31 August.

DANE (2015) *Learning about SBF models - Lesson 3.* Available at: http://dilab.cc.gatech.edu/dane/?page=LearningAboutSBFModels (Accessed: 23 Mar 2015).

Danilovic, M., & Browning, T. R. (2007) Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Management,* 25, pp. 300-314.

Davis, TP. (2006) Science, engineering, and statistics. *Applied Stochastic Models in Business and Industry*, 22 (5-6), pp. 401-430.

Dawson C. (2009) *Introduction to research methods: a practical guide for anyone undertaking a research project.* 5th edition. How To Books.

Deng, Y.M. (2002) Function and behaviour representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 16, pp.343-362.

Domb, E. and Kowalick, J. (1998) Applying TRIZ to develop new designs for the future of the airbag. *SAE Technical Paper*, 980647.

Dori, D. (2002) *Object-Process Methodology: a holistic systems paradigm.* Germany: Springer.

Dori, D. and Reinhartz-Berger, I. (2003) An OPM-Based Metamodel of System Development Process. *Lecture Notes in Computer Science: Proceedings of the 22nd International Conference on Conceptual Modelling*, 2813. Chicago, 13-16 October. Springer, pp. 105-117.

Dorst, K. and Vermaas, P.E. (2005) John Gero's Function-Behaviour-Structure Model of Designing: a Critical Analysis*. Research in Engineering Design*, 16(1-2), pp.17-26.

Durugbo, C., Tiwari, A. and Alcock, J.R. (2011) A review of information flow diagrammatic models for product-service systems. *International Journal of Advanced Manufacturing Technology*, 52, pp. 1193-1208.

Eckert, C. M., Stacey, M. K. and Clarkson, P. J. (2003) The spiral of applied research: A methodological view on integrated design research. *Proceedings of the 14th international conference on engineering design*, 19-21 August, Stockholm, Sweden.

Eckert, C.M., Alink, T. and Albers, A. (2010) Issue driven analysis of an existing product at different levels of abstraction. *Design Methods: Proceedings of the International Design Conference.* Dubrovnik, 17-20 May, pp.673-682.

Eckert, C. (2013) That which is not form: The practical challenges in using functional concepts in design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 27, pp. 217-231.

Eggert, R.J. (2005) *Engineering Design.* New Jersey: Pearson Prentice Hall.

Eisenbart, B., Qureshi, A., Gericke, K. and Blessing, L. (2013a) Integrating different functional modelling perspectives. *International Conference on Research into Design (ICoRD).* Indian Institute of Technology Madras, Chennai, 7-9 Jan.

Eisenbart, B., Gericke, K., Blessing, L. (2013b) Adapting the IFM Framework to Functional Approaches Across Disciplines. *Proceedings of International Conference on Engineering Design.* Sungkyunkwan University, Seoul, 19-22 Aug.

Eisenbart, B., Gericke, K. and Blessing, L. (2014) Application of the IFM framework for modelling and analysing system functionality. *International Design Conference (DESIGN).* Dubrovnik, 19-22 May, pp. 153-162.

Eisenbart, B. (2014) *Supporting interdisciplinary system development through integrated function modelling.* PhD dissertation. University of Luxembourg.

Eisenbart, B., Mandel, C., Gericke, K. and Blessing, L. (2015) Integrated function modelling: comparing the IFM framework with SysML. *Proceedings of the 20th International Conference on Engineering Design (ICED15).* Milan, 27-30 Jul.

Eppinger, S.D. and Browning, T.R. (2012) *Design structure matrix methods and applications.* Cambridge: the MIT press.

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

FAA (Federal Aviation Administration) (2006) *NAS system engineering manual.* FAA-ATO.

Fey, V. and Rivin, E. (2005) *Innovation on demand.* New York, Cambridge University Press.

Ford Design Institute (2004) *Failure mode and effect analysis - FMEA handbook (with robustness linkages).* Version 4.1. Dearborn: Ford Motor Company.

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

Friedenthal, S., Moore, A. and Steiner, R. (2012) *A practical guide to SysML: The Systems Modeling Language.* 2nd edition. Morgan Kaufmann.

Frezzini, F.R., Sachan, R. and Azimi, M. (2011) Review of systems engineering scope and processes, in Kamrani, A.K. and Azimi, M. (eds.) *Systems engineering tools and methods.* New York: CRC press, pp. 1-18.

Gadd, K. (2011) *TRIZ for engineers: enabling inventive problem solving.* John Wiley & Sons. Ltd.

Gero, J.S. (1990) Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4), pp .26-36.

Gero, J.S. and Kannengiesser, U. (2002) The situated function-behaviour-structure framework*. Artificial Intelligence in Design*, pp. 89-104.

Goel, A.K. and Stroulia, E. (1996) Functional device models and model-based diagnosis in adaptive design. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing (AI EDAM)*, 10, pp. 355-370.

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

Goel, A.K. (2013) A 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modeling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 27, pp.203-215.

Grix, J. (2004) *The foundations of research.* New York: Palgrave Macmillan.

Grobshtein, Y. and Dori D. (2011) Generating SysML views from an OPM model: design and evaluation. *Systems Engineering*, 14(3), pp. 327-340.

Hamraz, B., Caldwell, N.H.M., Wynn, D.C. and Clarkson, P.J. (2013) Requirements-based development of an improved engineering change management method. *Journal of Engineering Design*, 24 (11), pp. 765-793.

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

Harel, D. (1988) On visual formalisms. *Communications of the ACM*, 31(5), pp. 514-530.

Henshall, E., and Campean, F. (2009) *Implementing failure mode avoidance.* Society of Automotive Engineers Technical paper: 2009-01-0990.

Henshall, E., Campean, I., and Rutter, B. (2014) A Systems Approach to the Development and Use of FMEA in Complex Automotive Applications. *SAE Int. J. Mater. Manf.* 7(2) 280-290.

Henshall, E., Rutter, B., and Souch, D. (2015) Extending the Role of Interface Analysis within a Systems Engineering Approach to the Design of Robust and Reliable Automotive Product. *SAE Int. J. Mater. Manf.* 8(2) 322-335.

Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S. and Wood, K.L. (2002) *A functional basis for engineering design: reconciling and evolving previous efforts.* Washington: National Institute of Standards and Technology (NIST).

Hoffmann, H-P. (2011) *Systems engineering best practices with the rational solution for systems and software engineering.* IBM Corporation.

Howes, D.B. (2008) Conceptual modelling of human processes in a lunar systems architecture. *SAE Technical Paper*, 2008-01-2023. doi:10.4271/2008-01-2023.

INCOSE (2015) *INCOSE Systems Engineering Handbook: A guide for system life cycle processes and activities.* 4th edition. New Jersey: John Wiley & Sons.

International Organization for Standardization (2015) *ISO/PAS 19450:2015(en): Automation systems and integration - Object-Process Methodology.* https://www.iso.org/obp/ui/#iso:std:iso:pas:19450:ed-1:v1:en Accessed: 23 December 2015.

Jarratt, T.A.W. (2004) *A model-based approach to support the management of engineering change.* PhD thesis, University of Cambridge.

Johnson, C.W. (2005) What are emergent properties and how do they affect the engineering of complex systems? In Johnson, C. (editor) *Complexity in design and engineering.*, University of Glasgow, pp. 9-19.

Jupp, M.L., Campean, I.F. and Travcenko, J. (2013) Application of TRIZ to develop an in-service diagnostic system for a synchronous belt transmission for automotive application. *Procedia CIRP – Proceedings of the 2nd International Through-life Engineering Services Conference*, Elsevier.

Kaufman, J.J. and Woodhead, R. (2006) *Stimulating Innovation in Products and Services with Function Analysis and Mapping.* New Jersey: John Wiley & Sons.

Keller, R. (2007) *Predicting change propagation: algorithms, representations, software tools*. PhD thesis, University of Cambridge.

Keller, R., Alink, T., Pfeifer, C., Eckert, C. M., Clarkson, P. J., & Albers, A. (2007) Product models in design: A combined use of two models to assess change risks. *International Conference on International Design-ICED`07, Conference Proceedings*, 28th-31st Aug 2007, Paris.

Keuneke, A. M. (1991) Device representation - the significance of functional knowledge. *IEEE Expert*, 6(2), pp. 22-25.

Kim, J., Kim, I., Lee, H. and Park, B. (2012) A study on the role of TRIZ in DFSS. *SAE Int. J. Passeng. Cars - Mech. Syst.*, 5(1), pp. 22-29.

King, A.M. and Sivaloganathan (1998) Development of a methodology for using function analysis in flexible design strategies. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 212(3), pp. 215-230.

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*, 24, pp. 153-166.

Kitamura, Y. and Mizoguchi, R. (2010) Characterizing functions based on ontological models from an engineering point of view. *Proceedings of Formal Ontology in Information Systems (FOIS).* Amsterdam, pp. 301-314.

Komoto, H. and Tomiyama, T. (2011) A theory of decomposition in system architecting. *International Conference on Engineering Design*. Technical University of Denmark, 15-18 August.

Koopman, P.J. (1995) A taxonomy of decomposition strategies based on structures, behaviours and goals. *ASME Design Engineering Technical Conferences* , vol.2, pp. 611-618.

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

Kreimeyer, M. and Lindemann, U. (2011) *Complexity metrics in engineering design: managing the structure of design process.* Berlin: Springer-Verlag.

Kruchten, P. (2005) Casting Software Design in the Function-Behavior-Structure Framework. *IEEE Software*, 22(2), pp. 52-58.

Kurfman, M.A., Stone, R.B., Van Wie, M., Wood, K.L. and Otto, K.N. (2000) Theoretical underpinnings of functional modeling: preliminary experimental studies. *Proceedings of the ASME DETC.* Baltimore, 10-13 September.

Latronico, E. and Koopman, P. (2001) Representing embedded system sequence diagrams as a formal language. *UML*. Toronto Ontario, 3-5 October, pp. 302-316.

Lindemann, U., Maurer, M. &Braun, T. (2009) *Structural Complexity Management: An approach for the field of product design.* Berlin: Springer-Verlag.

Liu, C., Hildre, H.P., Zhang, H. and Rolvag, T. (2015) Conceptual design of multi-modal products. *Research in Engineering Design*, (26), pp. 219-234.

Long, J.E. (2000) Relationships between common graphical representations used in system engineering. *Proceedings of the SETE2000 Conference.*

Lough, K. G., Stone, R. and Tumer, I. Y.(2009) The risk in early design method. *Journal of Engineering Design*, 20(2), pp.155-173.

Lu, SC-Y. and Suh, NP. (2009) Complexity in design of technical systems. *CIRP Annals-Manufacturing Technology*, 58, pp. 157-160. DOI: 10.1016/j.cirp.2009.03.067

Mackenthun, R.,Voorhoeve, M. and Diagne, A. (2001) Methods. In: Girault, C. and Valk R., eds. *Petri nets for systems engineering: a guide to modelling, verification and applications.* Springer-Verlag, pp. 119-157.

Maier, M.W. and Rechtin, E. (2010) *The art of systems architecting.* 2nd edn. 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.

Maurer, M. and Lindemann, U. (2008) The application of the multiple-domain matrix. *IEEE international conference on system, man and cybernetics, SBC 2008.* pp. 2487-2493.

Mayer, R.J., Menzel, C.P., Painter, M.K., deWitte, P.S., Blinn, T. and Perakath, B. (1995) *IDEF3 process description capture method report.* Available at: http://www.idef.com/idef3.htm (Accessed: 19 August 2015).

McBride, R.T. (2005) *System analysis through bond graph modelling.* PhD thesis, University of Arizona.

Mital, A., Desai, A., Subramania, A. and Mital, A. (2008) *Product Development: a structured approach to consumer product development, design and manufacture.* Oxford: Butterworth-Heinemann.

Murata, T. (1989) Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4), pp. 541-580.

NASA (National Aeronautics and Space Administration) (2007) *Systems engineering handbook.* Rev1. Washington,D.C.

NIST (National Institute of Standards and Technology) (1993) *Integration definition for function modelling (IDEF0).*Federal Information Processing Standards (FIPS) Publication 183, NIST, Gaithersburg, MD, USA.

Nix, A.A., Sherrett, B. and Stone, R.B. (2011) A function based approach to TRIZ. *Proceedings of the ASME IDETC/CIE.* Washington, 29-31 August.

O`Donovan, B., Eckert, C., Clarkson J. and Browning, T.R. (2005) Design planning and modelling. In: Clarkson, J. and Eckert, C., eds. *Design process improvement: A review of current practice.* London: Springer-Verlag, pp.60-87.

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*, 14(4), pp. 364-382.

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

Pahl, G. and Beitz, W. (1988) *Engineering Design: a systematic approach.* Springer-Verlag.

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

Peleg, M. and Dori, D. (1999) Extending the object-process methodology to handle real-time systems. *Journal of Object-Oriented Programming*, 11(8), pp. 53-58.

Pimmler, T. U. (1994) *A Development methodology for product decomposition and integration.* MSc dissertation. Massachusetts Institute of Technology.

Pimmler, T.U. and Eppinger, S.D. (1994) Integration analysis of product decompositions. *ASME Design Theory and Methodology Conference*. Minneapolis, September, pp. 343-351.

Pineda, R.L. and Smith, E.D. (2011) Functional analysis and architecture, in Kamrani, A.K. and Azimi, M. (eds.) *Systems engineering tools and methods.* New York: CRC press, pp. 35-79.

Piques, J-D (2014) SysML for embedded automotive systems: SysCARS methodology. *Embedded Real Time Software and Systems (ERTS).* Toulouse, 05-07 February.

Pugh, S. (1991) *Total Design: integrated methods for successful product engineering.* Harlow: Addison-Wesley Publishers.

Roozenburg, N.F.M. & Eekels, J. (1995) *Product Design: Fundamentals and Methods.* Chichester: John Wiley &Sons Ltd.

Rovida, E., Bertoni, M. and Carulli, M. (2009) About the use of TRIZ for product-service development. *International Conference on Engineering Design (ICED).* Stanford, 24-27 Aug, pp. 133-144.

Santanam, C.B., Pope, J.E., Cheremisinoff, N.P. (1997) Heat transfer in Pope, J.E. (ed.) *Rules of thumb for mechanical engineers*. Texas: Gulf publishing company, pp. 18-50.

Saxena, A., Davis, T. and Jones, J.A (2015) A Failure Mode Avoidance Approach to Reliability. In *Reliability and Maintainability Symposium (RAMS)*. Palm Harbor, FL. 26-29 January. IEEE. pp. 1-6.

Shamieh, C. (2012) *Systems engineering for dummies.* Indiana: John Wiley & Sons.

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

Simon, H.A. (1996) *The sciences of the artificial. 3rd* edition. Boston: MIT Press.

Soderborg, N.R. (2002) *Representing systems through object-process methodology and axiomatic design.* MSc thesis, MIT.

Soderborg, N.R., Crawley, E.F. and Dori, D. (2002) System definition for axiomatic design aided by object-process methodology. *Proceedings of Second International Conference on Axiomatic Design - ICAD.* Cambridge, 10-12 June.

Srinivasan, V., Chakrabarti, A. and Lindemann, U. (2012) A framework for describing functions in design. *Design Methods: Proceedings of the 12th International Design Conference.* Dubrovnik, 21-24 May. Zagreb: the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, pp. 1111-1122.

Steward, D.V. (1981) The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transactions on Engineering Management*, 28(3), pp.71-74.

Stone, R.B. (1997) *Towards a theory of modular design.* PhD thesis. The University of Texas at Austin

Stone, R.B., & Wood, K.L. (2000) Development of a Functional Basis for design. *Journal of Mechanical Design*, 122, pp. 359-370.

Stone, R.B., Wood, K.L. and Crawford, R.H. (2000) A Heuristic method for identifying modules for product architectures. *Design Studies*, 21(1), pp. 5-31.

Suh, N.P. (1990) *The principles of design*. New York: Oxford University Press.

Suh, N. P. (2001) *Axiomatic design: advances and applications.* New York: Oxford University Press.

Suh, N.P. (2005) Complexity in Engineering. *CIRP Annals - Manufacturing Technology*, 54(2), pp.46-63.

Summers, J.D., Hernandez-Vargas, N., Zhao, Z., Shah, J.J., Lacroix, Z. (2001) Comparative study of representation structures for modelling function and behaviour of mechanical devices. *Proceedings of DETC*, Pennsylvania, 9-12 September.

Summers, J.D., Eckert, C. and Goel, A. (2013) Function in engineering: benchmarking representations and models. *Proceedings of International Conference on Engineering Design.* Sungkyunkwan University, Seoul, 19-22 Aug.

Tate, D. (1999) *A roadmap for decomposition: activities, theories, and tools for system design.* PhD thesis. Massachusetts Institute of Technology.

Tomiyama, T. and Yoshikawa, H. (1986) Extended general design theory. *Proceedings of the IFIP WG5.2 Working Conference.* Tokyo. North-Holland: Amsterdam, pp. 95-130.

Tomiyama, T., D`Amelio, V., Urbanic, J. and ElMaraghy, W. (2007) Complexity of multi-disciplinary design. *Annals of the CIRP,* 56(1), pp. 185-188. DOI:10.1016/j.cirp.2007.05.044

Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, Ch. and Kimura, F. (2009) Design methodologies: industrial and educational applications. *CIRP Annals - Manufacturing Technology*, 58, pp. 543-565.

Tomiyama, T., van Beek, T.J., Alvarez Cabrera, A.A., Komoto, H. and D'Amelio, V. (2013) Making function modeling practically usable. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 27(3), pp. 301-309. DOI: 10.1017/S0890060413000309.

Triengo, M. and Bos, A. (1985) Modeling the dynamics and kinematics of mechanical systems with multibond graphs. *Journal of Franklin Institute*, vol. 319, pp. 37-50.

Tumer, I.Y. and Stone R.B. (2001) Analytical method for mapping function to failure during high-risk component development. *Proceedings of ASME Design Engineering Technical Conferences (DETC'01)*. 9-12 September. Pittsburgh, Pennsylvania, USA.

Tumer, I.Y. and Stone, R.B. (2003) Mapping function to failure mode during component development. *Research in Engineering Design,* 14, pp. 25-33.

Tumer, I.Y., Stone, R.B. and Bell, D.G. (2003) Requirements for a failure mode taxonomy for use in conceptual design. *International Conference on Engineering Design.* Stockholm, 19-21 August.

Uddin, A., Campean, F. and Khan, M.K. (2015*)* Development of an interface analysis template for system design analysis*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15)*, Milan, Italy, 27-30 July.

Ullman, D.G. (1992) *The mechanical design process.* Boston: McGraw-Hill.

Ullman, D.G. (2010) *The mechanical design process.* 4thedn. New York: McGraw-Hill.

Ulrich, K. (1995) The role of product architecture in the manufacturing firm. *Research Policy*, 24, pp. 419-440.

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

Umeda, Y., Takeda, H., Tomiyama, T. and Yoshikawa, H. (1990) Function, behaviour, and structure, in Gero, J.S. (ed.) *Applications of Artificial Intelligence in Engineering V: Design.* Berlin: Springer-Verlag, pp. 177-193.

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

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

US DoD (2001) *Systems engineering fundamentals.* Virginia: Defense Acquisition University Press.

Van Beek, T.J.and Tomiyama, T. (2009) Integrating conventional system views with Function-Behaviour-State Modelling. *CIRP Design Conference.*

Vermaas P.E. (2007) The functional modelling account of Stone and Wood: some critical remarks. *International Conference on Engineering Design*, Paris, 28-31 August.

Vermaas P.E. (2008) On engineering meanings and representations of technical functions. *Proceedings of the ASME IDETC/CIE Conference*. New York, USA, 3-6 August.

Vermaas, P.E. (2009) The flexible meaning of function in engineering. *International Conference on Engineering Design.* Stanford University, Stanford, 24-27 August, pp. 113-124.

Vermaas, P.E. (2011) Accepting ambiguity of engineering functional descriptions. *Proceedings of the 18th International Conference on Engineering Design.* Technical University of Denmark, 15-18 August, pp. 98-107.

Vermaas, P.E. (2013) The coexistence of engineering meanings of function: four responses and their methodological implications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 27, pp. 191-202.

Weilkiens, T. (2006) *Systems Engineering with SysML/UML: Modelling, Analysis, Design.* Burlington: Morgan Kaufmann.

Wright, I.C. (1998) *Design methods in engineering and product design.* Berkshire: McGraw-Hill.

Wright, D. R. (2005) Finite State Machines. *CSC216*. Available at: http://www4.ncsu.edu/~drwrigh3/docs/courses/csc216/fsm-notes.pdf (Accessed: 19 August 2013).

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

Yang, K. and Zhang, H. (2000) A comparison of TRIZ and axiomatic design. *Proceedings of the ICAD.* Cambridge MA, 21-23 June, pp. 235-242.

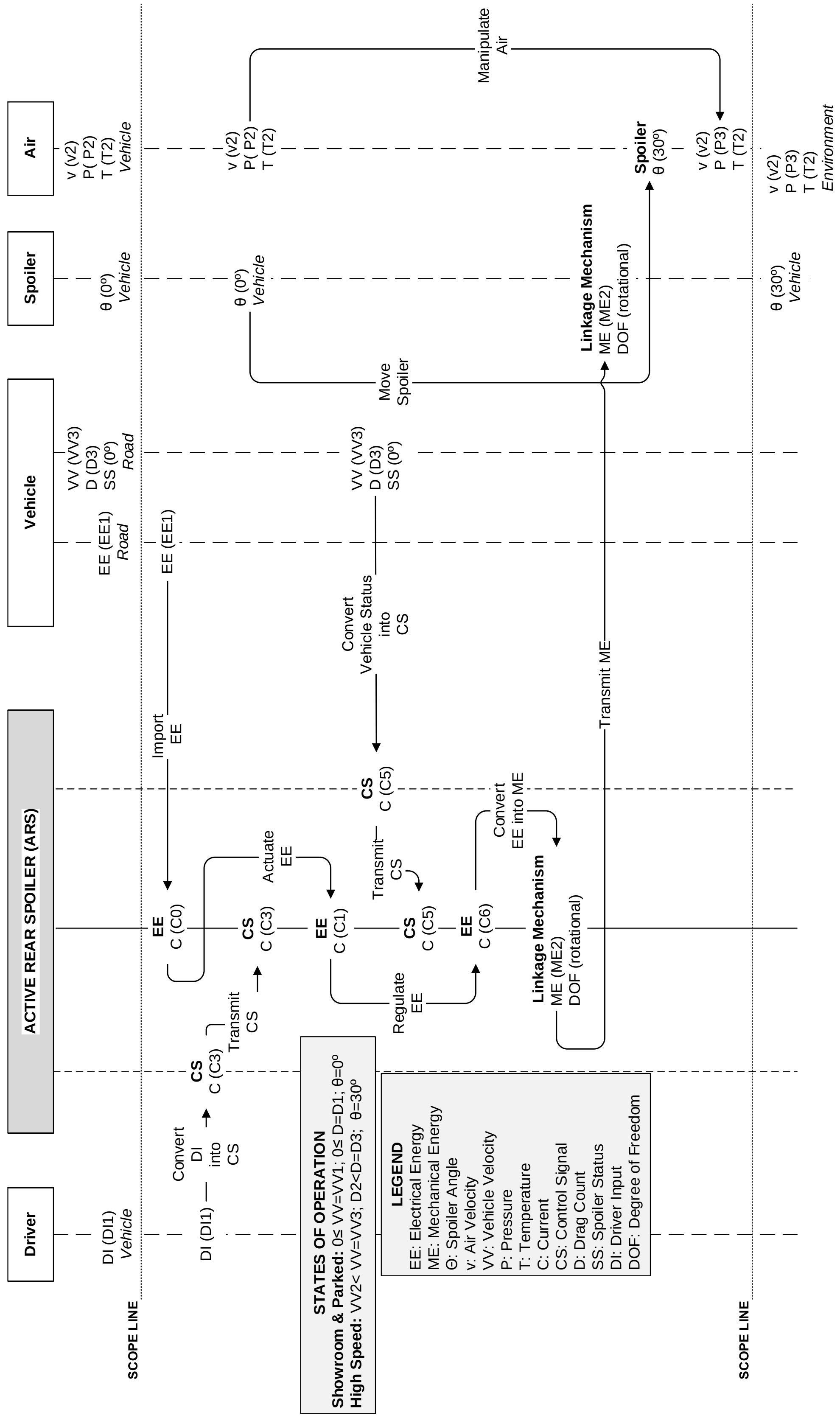
Yang, K. and El-Haik, B.S. (2009) *Design for six sigma: A roadmap for product development.* 2nd edn. USA: The McGraw-Hill Companies.

Yildirim, U. and Campean, I.F. (2013) An enhanced interface analysis method for engineering change management. *Smart product engineering: the 23th CIRP design conference.* Bochum, Germany, 11-13 March.

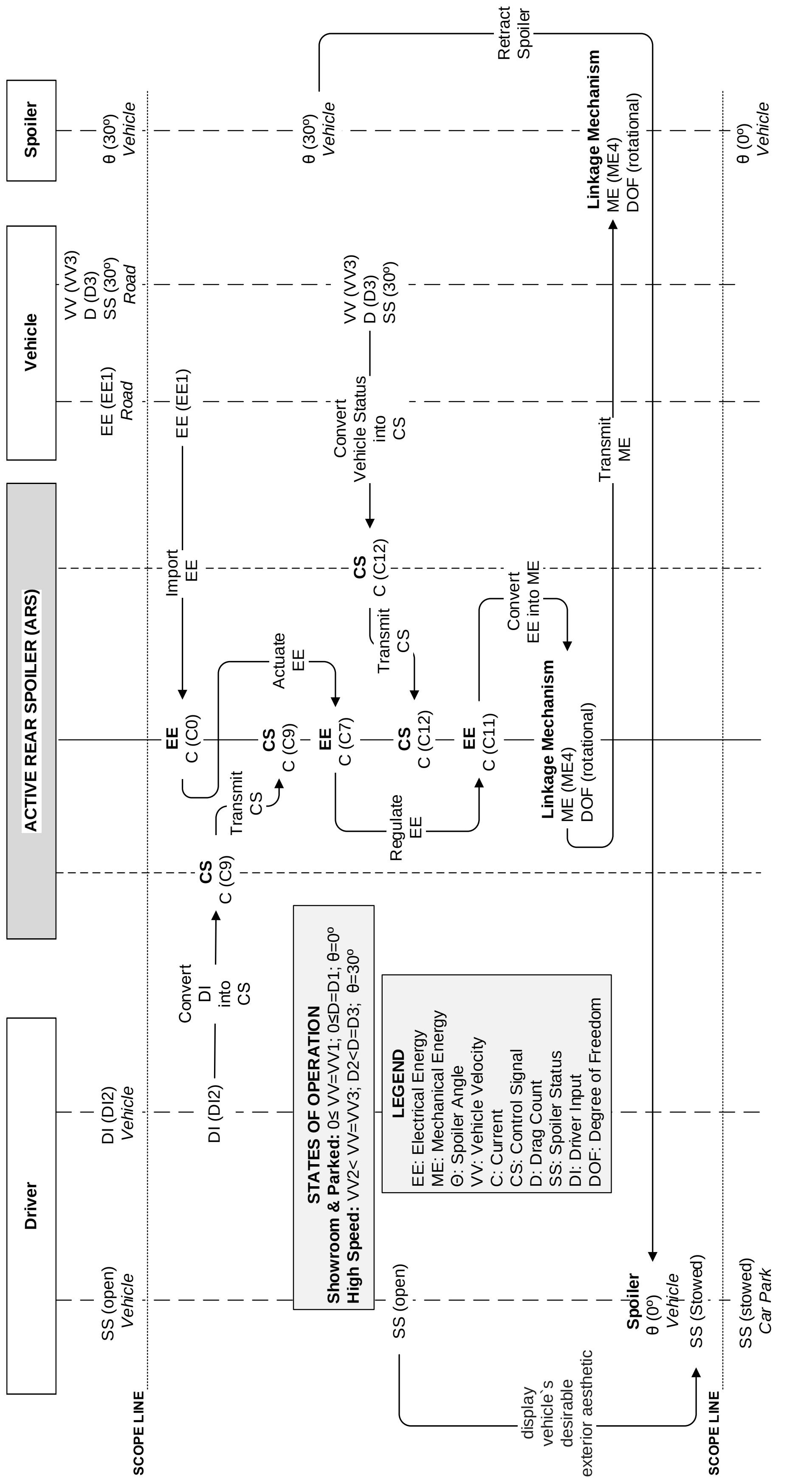
Zingel, C., Albers, A., Matthiesen, M. and Maletz, M. (2012) Experiences and advancements from one year of explorative application of an integrated model-based development technique using C&C^2-A in SysML. *International Journal of Computer Science (IJSC)*, 39 (2), pp.165-181.

Appendices

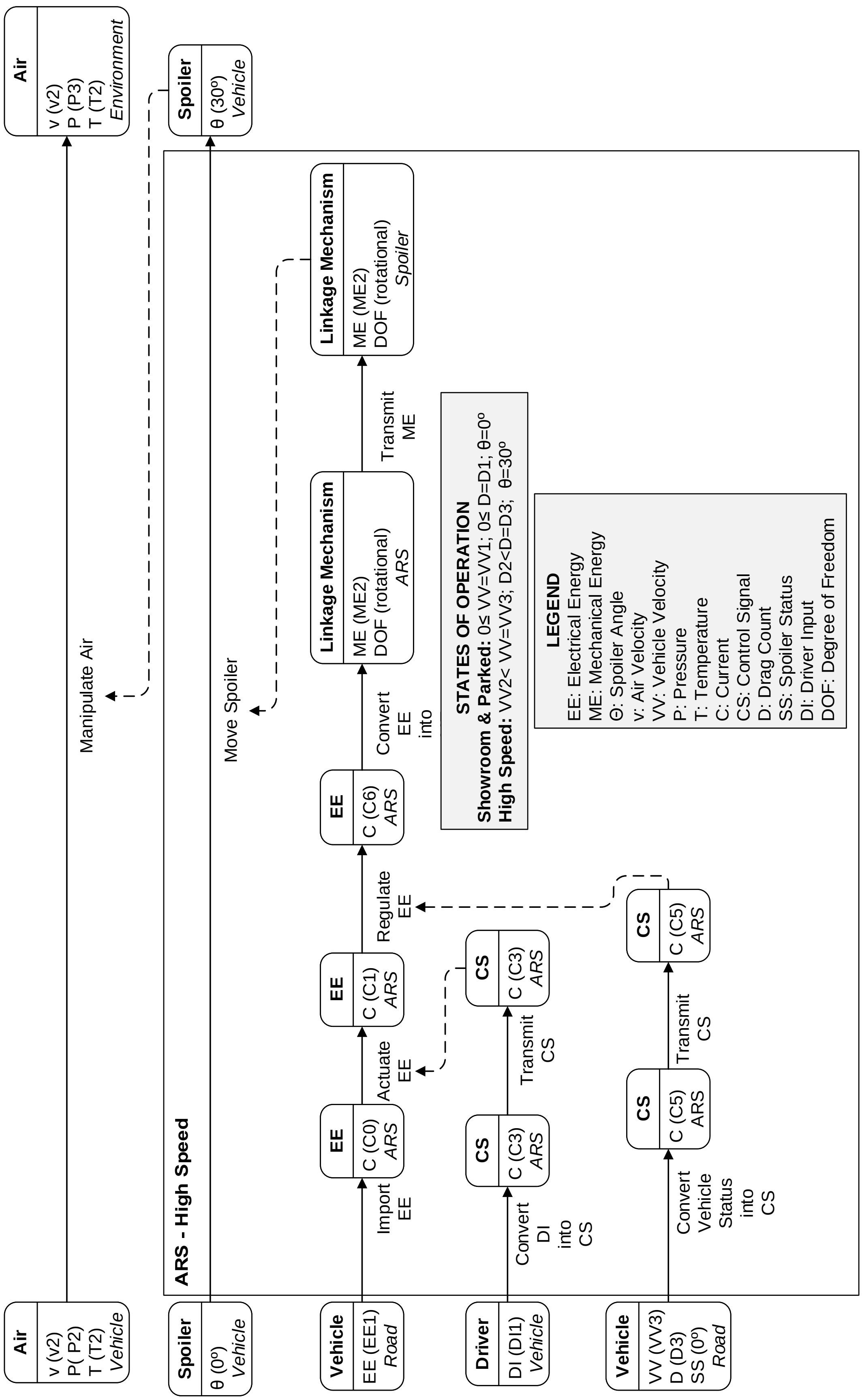
Appendix A: Sequence diagram for the use case “adjust spoiler angle in conjunction with the vehicle at high speed”



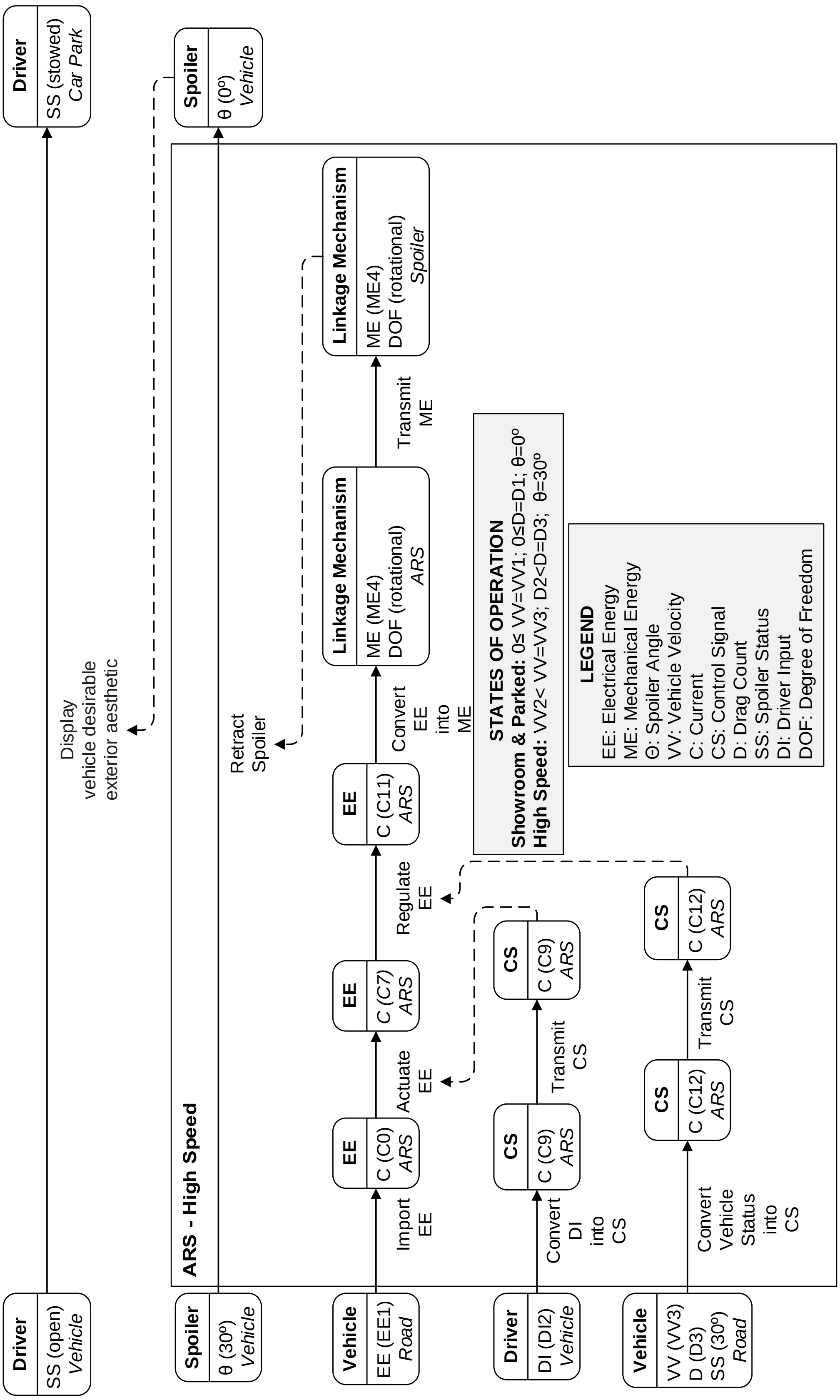
Appendix B: Sequence diagram for the use case “retract spoiler while the vehicle at high speed”



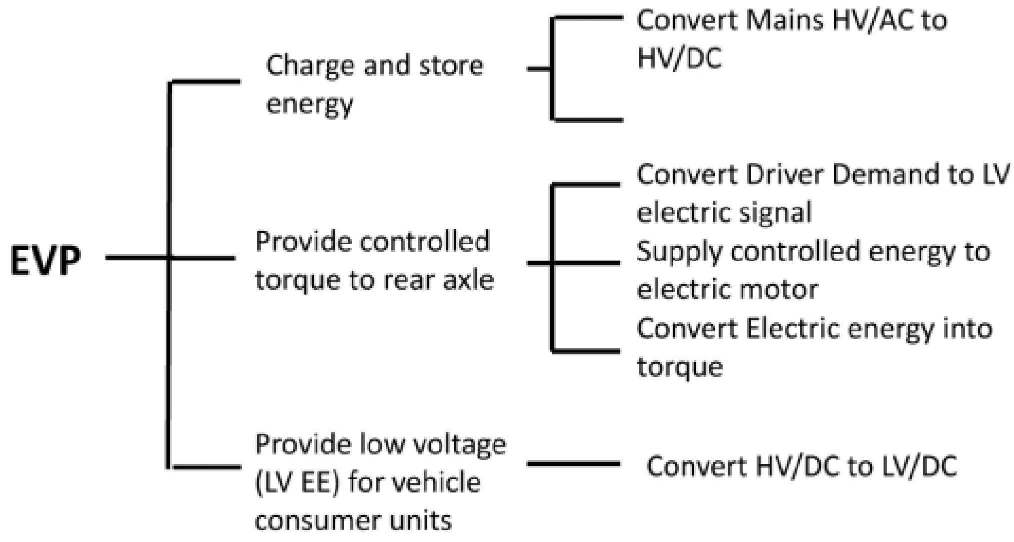
Appendix C: SSFD for the use case "adjust spoiler angle in conjunction with the vehicle at high speed"



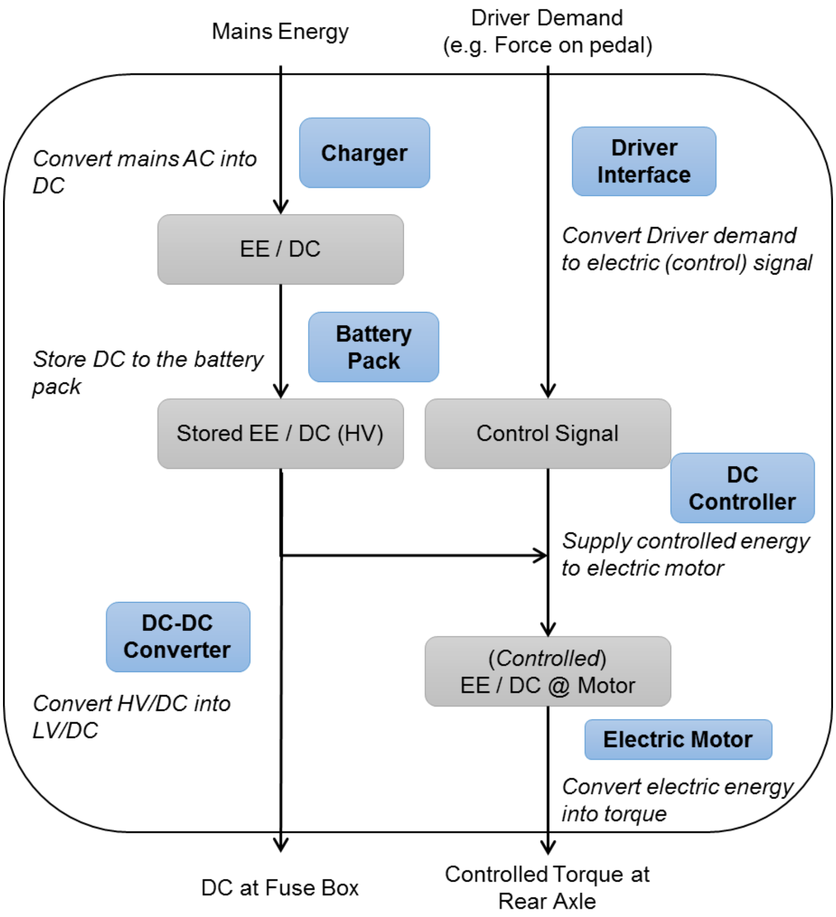
Appendix D: SSFD for the use case “retract spoiler while the vehicle at high speed”



Appendix E: Function Tree for the EV Powertrain (Campean et al., 2011)



Appendix F: System State Flow Diagram for the EV Powertrain (adapted from Campean and Henshall, 2012a, p.51)



Appendix G: Boundary Diagram for the EV Powertrain (Campean et al., 2011)