



THE DESCRIPTIONS AND ABSTRACTIONS OF A MULTIDISCIPLINARY SYSTEM: FROM GOALS TO STRUCTURE VIA COUPLING FRAMEWORK

A. Uddin

Abstract

In engineering design, there is a need for consensus to be developed on full descriptions and abstractions of a multidisciplinary system. The system's descriptions belong to its different abstractions. This paper introduces a structured coupling matrix (CM) framework that aids the practitioners to model and couple the system's descriptions from goals to structure via functions at its abstractions from black-box to white-box. The CM framework is built upon limitations in existing frameworks and is illustrated with an example. It provides structured guidelines for system analysis and design.

Keywords: system descriptions, system abstractions, design methods, engineering design, systems engineering (SE)

1. Introduction

Due to increasing number of requirements with the introduction of new features and technologies into current multidisciplinary system development projects, designers from different engineering disciplines need to work in close collaboration to develop comprehensive understanding on the design ambition. This requires fixation and agreement on two fundamental design information aspects among multidisciplinary engineers at conceptual design stage: (1) the full technical *descriptions* of a system and (2) their extractions at relevant *abstractions* of a system in increasing order of detail. In engineering design, Vermaas (2009) presented five layers necessary for a full description of a technical system as shown in Figure 1a whilst Acharya and Pandya (2012) presented three key abstractions necessary for testing complex systems as shown in Figure 1b. In existing research, there are insufficient guidelines that describe and explore the relation between descriptions and abstractions of a multidisciplinary system as depicted in Figure 2. To resolve this issue, this paper presents a structured approach.

The top-down reasoning for a system analysis (as per Vermaas's model in Figure 1a) begins with the identification of the *goals* of the technical system with its stakeholders (or external elements in its environment) and their associated possible *actions (i.e. operations)* with it, in relation to specific use plan (i.e. scenario). A plan or scenario consists of executable operations which are ordered sequentially or otherwise. The *actions or operations* that exist between technical system and its environment lead to *behavioural constraints* that subsequently help in deriving *functions*. According to Vermaas (2009) *functions* describe the roles the technical system should play in its environment. *Behaviour* describes the way in which the physicochemical state of the system evolves in its environment. Finally, the solution (*structure*) need to be explored in a way that the configured internal elements of a system as a whole should exhibit the behavioural interactions captured with its external environment and perform the defined functions. The function concept is further split into environment-centric functions and device-centric functions (Chandrasekaran and Josephson, 2000; Vermaas, 2009; Crilly, 2012). According to Crilly

(2012), *environment-centric functions* need to only refer to elements external to the system whilst *device-centric functions* need only refer to system's specific (internal) structural elements. Even though, the five-layer conceptual model can effectively guide multidisciplinary engineers to understand and agree on what technical descriptions should be defined and analysed for describing a system, it does not provide guidelines on system abstraction views that are often analysed, and debated among the mechanical, electrical, control, and software engineers during the system testing and development as shown in Figure 1b.

Mechanical engineers often start a design task of a system by referring it black-box as a mean to represent its higher level (see e.g. Pahl et al., 2007; Kossiakoff et al., 2011) in the context of defining its intended functionality with its surrounding environment whereas software testing engineers use same term of black-box in the context of testing the system's functionality (see e.g. Alexander and Zink, 2002). There are two more terms: grey-box and white-box, generally used by test engineers in software testing. **Black-box** testing requires supply of valid set of inputs by tester to test the system's use case and to examine the resulting expected outputs without looking into its internal functionality or implementation details (Alexander and Zink, 2002; Rational Software, 2003; Acharya and Pandya, 2012). In contrast, a **white-box** test requires the tester to look inside the box or implementation details (Alexander and Zink, 2002; Rational Software, 2003). The black-box term is very common in both systems engineering (see e.g. Rational Software, 2003; Kossiakoff et al., 2011) and engineering design communities (see e.g. Otto and Wood, 2001; Pahl et al., 2007) in contrast to white-box term. The **grey-box** testing is a combination of both black-box and white-box details and it acts as a bridge in between them (Acharya and Pandya, 2012). In existing state of art, there exists no approach or framework that could integrate and map system's (1) full descriptions at its (2) relevant abstraction views as shown in Figure 2. This paper introduces a Coupling Matrix (CM) framework that aids multidisciplinary engineers with structured guidelines to define and integrate the descriptions of the design at relevant abstractions of a system.

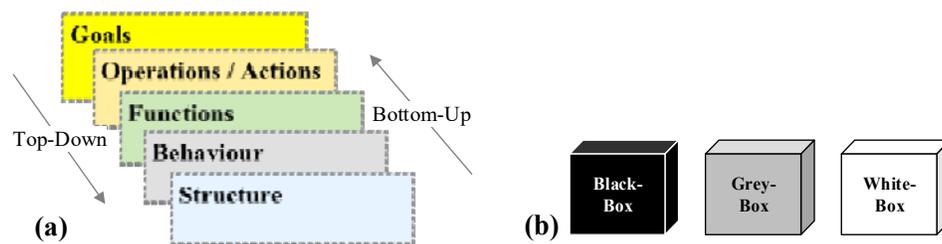


Figure 1. Descriptions model and abstractions model of a multidisciplinary system (adapted from Vermaas, 2009; Acharya and Pandya, 2012)

		Descriptions for a Multidisciplinary System				
		Goals	Actions	Functions	Behaviour	Structure
Abstractions for a Multidisciplinary System	Black-Box	No implementation information to Implementation information				
	Grey-Box					
	White-box					

Figure 2. Scope for needed framework to trace and integrate the design aspects

The remainder of the paper is structured as follows: Section 2 briefly reviews the limitations of existing system analysis approaches, yielding to scope for a needed framework. Section 3 briefly introduces the approach of goal-to-structure CM framework followed by its illustration with an example of ink-pen case study. Section 4 discusses the strengths and weaknesses of CM framework including the findings gained during its application on real world case studies in an automotive company. Section 5 concludes the paper.

2. Literature review

In this section, the aim of the developed CM framework is highlighted by reviewing the limitations of existing system analysis or modelling frameworks. At the same time, the baseline descriptions recommended by the current frameworks for system analysis and design are incorporated.

Quality function deployment (Hauser and Clausing, 2009) is the comprehensive framework that maps the two descriptions; customer goals or needs into *performance specifications* and thereafter the

performance specifications to a system's implementation structure i.e. internal elements of a system. Though the approach is quite robust, the definition of the functions over the goals, performance specifications, and implementation structure is not discussed whereas the developed CM framework aims to couple the goals, performance specifications to functions (i.e. both environment-centric and device-centric descriptions) and then their allocation to system's implementation structure.

Axiomatic design (Suh, 2001) provides design matrix as an approach that maps two descriptions; the functional requirements of a system to design parameters and follows two axioms i.e. (i) maintain the independence of functional requirements (ii) minimise the information content. Though the axioms and design matrix provide a robust approach for evaluating design decisions but the design parameters describe the chosen implementation that satisfy functional requirements whereas the proposed CM framework supports coupling of the goals, environment-centric functions, and performance specifications to device-centric functions without implementation in mind. The guidelines on distinction of functional requirements of a system in terms of its environment-centric and device centric is also not recognised or supported by design matrix.

Bonnema (2008) provides FunKey architecting framework where stakeholders' performance specifications on a system are coupled with functions without the system implementation in mind. It also aids designers to think of budgets distribution across system functions. Though, FunKey approach provides intuitive thinking; it does not suggest or support other descriptions such as goals, and actions (or operations) required for system analysis.

Eisenbart (2014) provides integrated function modelling (IFM) framework based on series of matrices where each matrix maps the technical process description with other system descriptions such as use cases (the goals), structural elements, and *states*. The process description implicitly covers both function and behaviour descriptions. In IFM, technical processes are mapped to goals, implementation structure, and states of both *flows* (energy, material, and information) and structural elements. Though, the IFM is a very comprehensive approach where various descriptions within functional modelling domain are integrated in a structured manner; it does not draw the distinctions explicitly between actions (or operations), environment-centric, and device-centric descriptions.

Thus, it is seen that, on one hand, existing matrix-based frameworks do by-pass some essential descriptions of a system when benchmarked against five-layer conceptual model (Figure 1a). Similarly, on the other hand, these frameworks also describe other descriptions like performance specifications, flows, and states which are not explicitly discussed in the five-layer conceptual model. It is also observed that not enough guidelines are discussed on maintaining system abstraction views (Figure 1b) in relation to such descriptions by the existing frameworks and the conceptual model. In design practice, multidisciplinary engineers often struggle to describe incomplete descriptions (see e.g. Eisenbart, 2014) and to scope out the system abstractions which results in an insufficient understanding of design ambition. It is presented in this paper that both aspects (i.e. *descriptions* and *abstractions*) are essential and thus demands for a needed framework that could facilitate structured guidelines for multidisciplinary systems' analysis and design.

3. Introduction to CM framework

The CM framework intends to aid multidisciplinary engineers to analyse and integrate the system descriptions from goals to structure at relevant system abstractions by providing a cohesive overview of system analysis and design. Figure 3 shows the concepts of CM framework. It consists of concepts mainly equivalent to five-layer conceptual model (each colour in Figure 3 represents same layer as in Figure 1a). The descriptions such as goals, actions, and environment-centric functions, flows, and performance specifications are defined at system's black-box which refers only elements external to system. The device-centric functions are described at system's grey-box with same input-output flows (identified at black-box) with their intermediate states defined across logical paths existing in between them. The descriptions of black-box and grey-box are then assigned to implementation structure that refers to elements internal to system at white-box. At white-box, there can be more than one structural configurations possible to meet the descriptions defined and analysed at black-box and grey-box abstractions. The interdependencies of descriptions from system's black-box to white-box abstractions are mapped through four matrices as shown in Figure 3.

3.1. Approach

In the first step of CM approach, the system is treated as a black-box and the goals of it are determined with respect to its external involved elements (e.g. user and other interacting elements) that may have got some vested interest in the system. This step helps in defining the system's context around specific goal(s) and eliciting the system's interfaces in its external environment. In the next step, the expected interaction operations/actions between system and its external elements are determined sequentially (or non-sequentially). Subsequently, the input and output flows as behavioural constraints of the system are extracted that can occur within the interaction operations. These behavioural flows in turn support in deriving environment-centric functions of the system associated with its external elements to accomplish the goals. The performance specifications of environment-centric functions are specified based on input-output flows. The interdependencies between goals and environment-centric functions are explored at this stage. Once the black-box of the system is detailed from goals to environment-centric functions, it is then decomposed into grey-box with same input-output flows that subsequently help in determining the device-centric functions based on intermediate transitions of flows from their input to output states. The device-centric functions are the means by which environment-centric functions are achieved and thus mapping between these two function types is performed at this stage of system analysis (i.e. after the grey-box establishment). The mapped functions (from environment-centric to device-centric) are then analysed to search for structural elements (i.e. internals) of the system. There can be more than one structural implementations possible. The structural elements are configured at white-box in a way that satisfy both grey-box and black-box descriptions.

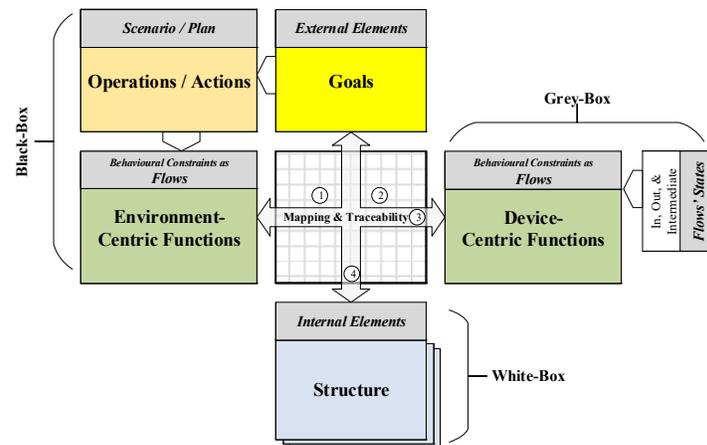


Figure 3. The Coupling Matrix (CM) framework

3.2. Example: Ink-pen

A literature review example i.e. ink-pen device is analysed to describe the approach of CM framework. The ink-pen case study has been used by many researchers (King and Sivaloganathan, 1999; Van Wie et al., 2003; Brown and Blessing, 2005; Vermaas, 2009; Crilly, 2012) to reason about descriptions and abstractions of a technical system.

3.2.1. Descriptions at black-box

Goals (and external elements)

Figure 4-a describes two generic goals for the ink-pen. Each goal of the ink-pen is associated with its relevant external elements that are involved in its accomplishment. A goal "G1: to provide information to another person" is analysed with respect to its external environment or elements involve in it i.e. user, reader, and paper. There can be many other goals of an ink-pen that can determine other external elements. For example, "G2: to carry through" that would exist between an ink-pen, user, and transporter. The two different goals of a system may involve a common external element e.g. user is involved in both goals G1 and G2.

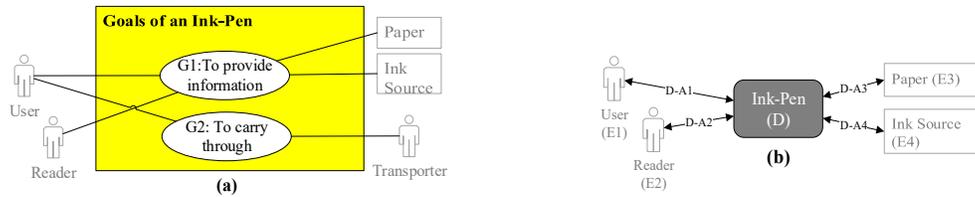


Figure 4. Use cases and the context diagram for goal "provide information"

Once the goal(s) and relevant associated external elements are identified then one can state that system external interfaces with its environment are established at black-box as discussed by Uddin et al. (2016) using use case diagram (Figure 4-a) and context diagrams (Figure 4-b). For example, in Figure 4-b, for the goal-G1 of an ink-pen device (D), there are 4-interfaces identified (i.e. D-E1, D-E2, D-E3, D-E4) with its external environment being user (E1), paper (E3), ink-source (E4), and reader (E2). Once the goals and interfaces of an ink-pen are identified, the next step is to analyse (the sequential or non-sequential) operations that should happen between an ink-pen and its external interfaces.

Interaction operations/actions (and behavioural effects)

There can be many sequential (or non-sequential) interaction operations in order to achieve a goal of an ink-pen. Figure 5 shows that the ink-pen (at its black-box) interacts directly with user, paper, and ink source to accomplish the goal "G1: provide information" via interaction operations in order of "O2: grips the pen", "O3: orients the pen", till "O6: conveys the information". The interactions can be modelled via sequence diagram as discussed by Satzinger et al. (2012). Reader is indirectly involved with the ink-pen device whereas other external elements directly interface with it.

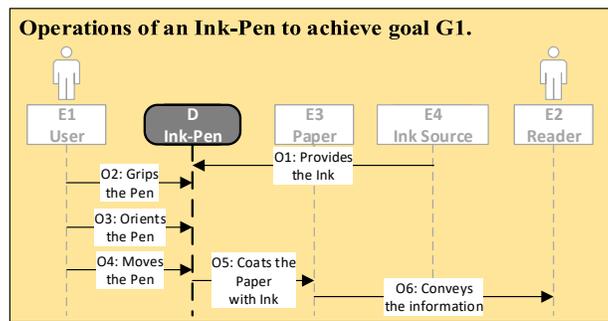


Figure 5. Interaction operations for goal G1- provide information at black-box

The interaction operations are then analysed to identify the expected behavioural effects as incoming and outgoing **flows** from/to an ink-pen system. At this stage of analysis, the designers would initially analyse **desired** behavioural effects as flows that should happen within the causal interaction operations in the main success plan or scenario. For example, shown in Figure 6, "ink" (i.e. a material flow) should flow from ink-source to ink-pen which would happen within the interaction operation "O1: provides the ink". Also, the "user's effort or force" flow from user to ink-pen would occur within the interaction

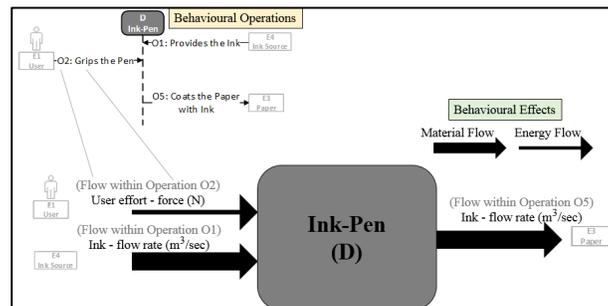


Figure 6. Abstraction of input-output flows of an ink-pen at black-box

operation "O2: grips the pen". Similarly, all interaction operations can be analysed to extract relevant set of desired behavioural effects as flows in terms of inputs and outputs to/from ink-pen at its black-box on its surroundings or external elements. A flow can have measurable performance metrics as discussed in (Hirtz et al., 2002; Lucero et al., 2016). Figure 6 shows performance metrics "flow rate-m3/sec" of an "ink" flow and "force in newton-N" of "user effort's" flow. These metrics support later in specifying the performance specifications to the functions of an ink-pen.

Environment-centric functions (and performance specifications)

Brown and Blessing (2005) describe environment-centric function of pen as: "to cause a piece of paper to have ink on it". This descriptive statement holds true at black-box of an ink-pen (in Figure 6) as external element (paper) is affected by device (pen) having operand/flow (ink as out) and thus environment centric function is between device and external element. However, the 'verb' in above environment-centric function is not clear and thus better way of articulating it from pen device perspective would be; "export ink from ink-pen (i.e. device) to paper (i.e. external element)" as shown in Figure 7. According to Crilly (2012), *environment-centric functions* need to only refer to elements external to the system. Similarly, the other environment-centric functions can be identified with ink-pen's external interfaces based on input-output flows that are in turn identified and analysed within interaction operations O1 to O5.

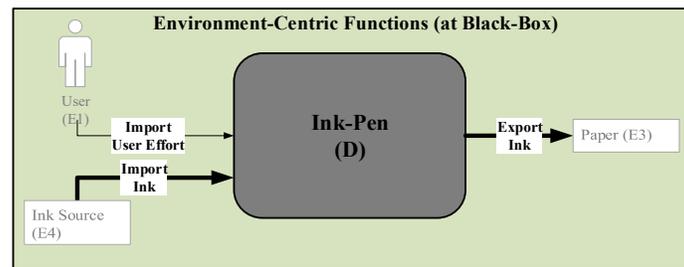


Figure 7. Environment-centric functions of an ink-pen at black-box

All such descriptions from goals to environment-centric at system's black-box can be identified and specified textually via interface analysis template (IAT), developed by Uddin et al. (2016). Figure 8 represents the technical descriptions derived in increasing order of detail from column S1 to S6. The textual template in Figure 8 represents an accumulated view of an ink-pen's analysis from Figure 4 to Figure 7. Based on flows' properties, the performance specifications associated with each function is also specified after listing functions. For example, shown in Figure 8 in column S6, the function 'import ink' in D-E4 interface and 'export ink' in D-E3 interface would have same performance metric i.e. 'ink flow rate' but different performance values i.e. X (being minimum performance value) and Y (being maximum performance value) for a goal "G1: provide information".

Step 2: Interface	Step 3: Operation Description	S4: Exchanges / Flows Description				S6: System Requirements Description			S1: Goals Description		
System External Environment	Description of Interaction Operations	Flow Type	Exchange / Flow	From	To	Environment-Centric Functions			Performance Target / Measure	G1: Provide Information	G2: Carry Through
						Verb	Object/Noun	Input / Output			
Ink-Pen - User Interface (D-E1)	Main Plan / Scenario										
	O2: User grips the Pen	E	User Effort	User	Ink-Pen	Import	User Effort		User Force (X < kN < Y)	X	X
	O3: User orients the Pen	①
Paper - Ink-Pen Interface (D-E3)	O4: User moves the Pen
	O5: Pen coats the Paper	M	Ink	Ink-Pen	Paper	Export	Ink		Ink flow rate (X < m3/sec < Y)	X	
Ink-Source - Ink-Pen Interface (D-E4)	O1: Ink Source provides the ink	M	Ink	Ink-Source	Ink-Pen	Import	Ink		Ink flow rate (X < m3/sec < Y)	X	

Figure 8. Goals vs environment-centric functions coupling at black-box

Mapping of environment-centric functions to goals descriptions via 1st matrix:

At this stage of system analysis, the derived environment-centric functions (Figure 7) during the analysis of a specific goal may belong to other goals (Figure 4-a) of a system and thus first mapping step of CM framework occurs between these two descriptions. For example, the derived environment-centric function "import user effort" (in column S6 of Figure 8) is also mapped across goal "G2: to carry through" though initially it was derived in the context of goal "G1: to provide information to another person" (in column S1 of Figure 8).

It should also be noted that the performance specification of a same function can vary from one goal to another goal which can be captured in Figure 8. For example, the target value of a performance metric i.e. user effort " $X < kN < Y$ " of environment-centric function "import user effort" for a goal "G2: carry through" may be different from that of goal "G1: provide information" during writing. In Figure 8, it is assumed that performance value would be same across both goals.

3.2.2. Descriptions at grey-box

Device-centric functions (and flows' intermediate states)

The input-output flows in Figure 6 of an ink-pen are then used to determine the device-centric functions based on intermediate transitions of main flow of "ink" from its initial state to final state. Many functional modelling techniques as available in (Otto and Wood, 2001; Hirtz et al., 2002; Yildirim and Campean, 2014), can be used and adapted for such purpose. For example, Figure 9 shows the device-centric functions "store ink", "transport ink", and "disperse ink" based on main ink flow but without its states' and attributes details. Figure 10 shows the device-centric functions based on ink flow's states' transitions with measurable attributes.

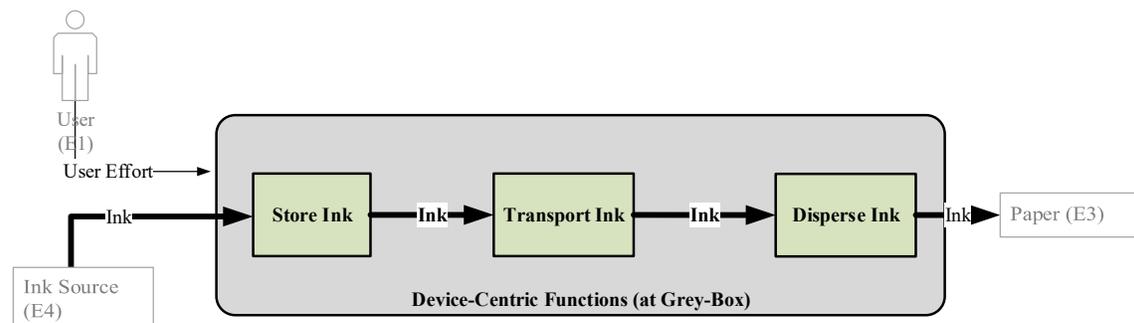


Figure 9. Device-centric functions of an ink-pen

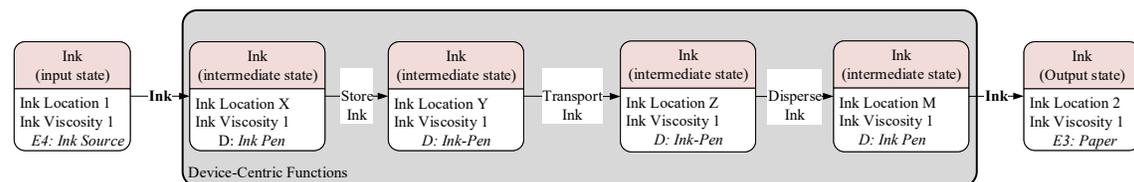


Figure 10. Device-centric functions with flows' states representation

Mapping of device-centric descriptions to environment-centric descriptions via 2nd and 3rd matrix:

The device-centric functions "store ink", "transport ink" (Figure 9) of ink-pen are the means by which environment-centric function "import ink" (Figures 7 & 8) and the goal "G1: provide information" (Figure 4-a) are accomplished. Thus, the interdependencies or mapping in between them should be established at this stage of system analysis. This can be accomplished using matrix "goals vs device-centric functions" as shown in Figure 11 and also for "environment-centric vs device centric functions" in Figure 12. The environment-centric descriptions are specified in from Figure 8. The mapping order of device-centric functions across goals and environment-centric functions can differ but the important is to recognise and establish links in between these three descriptions.

Goals (G)	G2: Carry Through			②	X
	G1: Provide Information	X	X	X	X
System Black-Box		System Grey-Box			
Device-Centric Functions					
Store Ink	Transport Ink	Disperse Ink	Sustain User Effort		

Figure 11. Goals vs device-centric functions coupling from black-box to grey-box

Interface Description	Operation Description	Exchanges / Flows Description	System Black-Box				System Grey-Box						
			Flow Type	Exchange / Flow	From	To	Environment-Centric Functions		Performance Target / Measure	Device-Centric Functions			
							Verb	Object/Noun		Store Ink	Transport Ink	Disperse Ink	Sustain User Effort
System External Environment	Description of Interaction Operations (O)												
	Main Plan / Scenario												
Ink-Pen - User Interface (D-E1)	O2: User grips the Pen	E	User Effort	User	Ink-Pen	Import	User Effort	User Force ($X < kN < Y$)					X
	O3: User orients the Pen			③		
	O4: User moves the Pen					
Paper - Ink-Pen Interface (D-E3)	O5: Pen coats the Paper with Ink	M	Ink	Ink-Pen	Paper	Export	Ink	Ink flow rate ($X < m^3/sec < Y$)			X		
Ink-Source - Ink-Pen Interface (D-E4)	O1: Ink Source provides the ink	M	Ink	Ink-Source	Ink-Pen	Import	Ink	Ink flow rate ($X < m^3/sec < Y$)	X	X			

Figure 12. Device-centric vs environment-centric functions coupling from black-box to grey-box

It should be noted that in current state of art there exists as such no functional modelling method that could explicitly draw the distinction and represent both environment-centric and device-centric functions in a single diagram. To substantiate this point, consider Figure 13 that shows a combined functional view of an ink-pen's environment-centric functions (consistent with Figure 7) and device-centric functions (consistent with Figure 9).

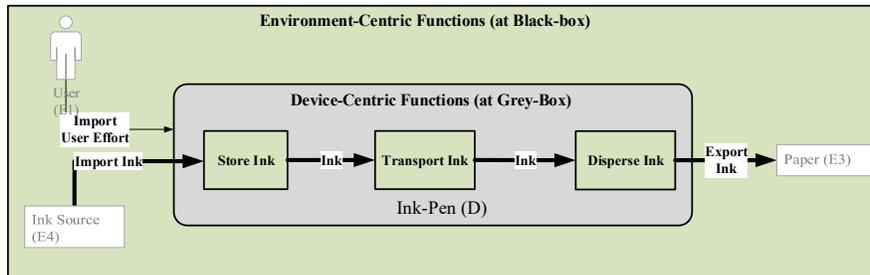


Figure 13. Combined view of environment-centric and device-centric functions

3.2.3. Descriptions at white-box

Structural elements:

On the basis of environment-centric (at black-box) and device-centric descriptions (at grey-box) and having established the relationships in between them in Figures 11 and 12, the structural solutions or implementations are explored at white-box of an ink-pen system that would deliver its required goals, environment-centric, and device-centric functions. Two structural implementations of an ink-pen are possible as shown in Figure 14. Figure 14 represents the whole template of CM framework. At this stage of system analysis, the required descriptions at system level's black and grey boxes are cascaded to its internal elements. For example, the device centric function "store ink", and "transport ink" and the environment-centric function "import ink" are allocated/cascaded to an internal element "A1: ink-container" as shown in Figure 14.

The most suitable structural configuration based on design trade-off criteria such as cost, and manufacturing time, including functional specifications can then be taken forward for further embodiment design. For example, assuming that the structural configuration no. 1 of an ink pen in Figure

14 is chosen for development purpose, its system boundary diagram is then developed for further in depth behavioural analysis as shown in Figure 15.

Interface Description	Operation Description	Exchanges / Flows Description		System Black-Box			System Grey-Box					
				Environment-Centric Functions		Performance Target / Measure	Device-Centric Functions					
System External Environment	Description of Interaction Operations (O)	Flow Type	Exchange / Type Flow	From	To	Verb	Object/Noun	Output	Store Ink	Transport Ink	Disperse Ink	Sustain User Effort
	Main Plan / Scenario											
Ink-Pen - User Interface (D-E1)	O2: User grips the Pen O3: User orients the Pen O4: User moves the Pen	E	User Effort	User	Ink-Pen	Import	User Effort					X
Paper - Ink-Pen Interface (D-E3)	O5: Pen coats the Paper with Ink	M	Ink	Ink-Pen	Paper	Export	Ink				X	
Ink-Source - Ink-Pen Interface (D-E4)	O1: Ink Source provides the Ink	M	Ink	Ink-Source	Ink-Pen	Import	Ink		X	X		

System White-Box	Structural Elements Configuration No. 1
X	X
	④
	X
	X
	...
System White-Box	Structural Elements Configuration No. 2
X	X
	X
	X
	X
	X

Figure 14. An excerpt of an ink-pen analysis via CM framework

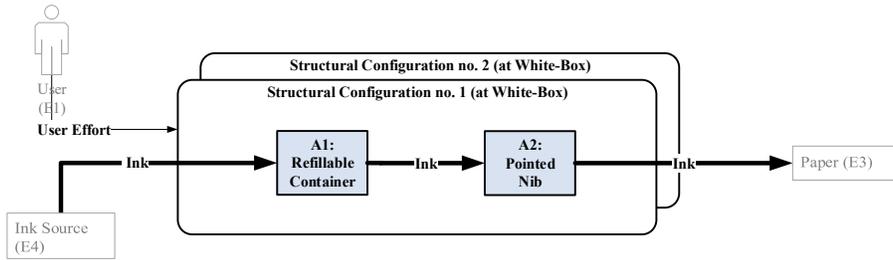


Figure 15. An excerpt of an ink-pen's system boundary diagram

Once the system boundary diagram is developed, then the interface analysis in between chosen elements can be conducted via interface analysis template (IAT) as shown in Figure 16. This would in turn identify the environment-centric descriptions for each of the internal design elements. For example, from ink-container's perspective, it would be "export ink" to nib whereas from nib's perspective, it would be "import ink" from ink-container.

S2: Interface	S3: Operation Description	S4: Exchanges / Flows Description			S6: System Requirements Description		
System External Environment	Description of Interaction Operations	Flow Type	Exchange / Type Flow	From	To	Interaction Functions	Performance Target / Measure
						Verb	Object/Noun
Refillable Container - Pointed Nib Interface (A1 - A2)	O1: Manage the ink flow	E	Ink	Container	Nib	Export	Ink
Pointed Nib - Refillable Container (A2 - A1)	O1: Manage the ink flow	M	Ink	Container	Nib	Import	Ink

Figure 16. Interface analysis between internal elements of an ink-pen

All such descriptions between nib and container are device-centric descriptions for their system i.e. an ink-pen. According to Crilly (2012) device-centric functions need only refer to system's specific (internal) structural elements. However, these descriptions in turn at the white-box of an ink-pen become environment-centric descriptions for each of its internal elements because each element (e.g. container) on its own can now be treated as a black-box as can be visualised in Figure 15. The CM framework in Figure 14 provides the integrated view of ink-pen analysis with the establishment of traceability links across the various descriptions from its goals (Figure 4-a), operations at its interfaces including environment-centric functions (Figures 7 and 8) and device-centric functions (Figure 9) as well as the possible implementation structures (Figure 15).

4. Discussion

One of the key findings of the literature review in Section 2 towards describing the system's descriptions is that the existing matrix-based frameworks are incomplete and ambiguous against the five-layer conceptual reasoning model (i.e. *goals to structure*). However, at the same time, these frameworks offer other descriptions (such as *performance specifications, flows, and states*) that are not explicitly discussed in five-layer conceptual model. On one hand, the five layer conceptual model suggests WHAT needs to be described in top-down for a technical system analysis. In contrast to this, the existing frameworks suggest HOW to manage different system's descriptions. However, both do not explicitly discuss the guidelines that WHAT descriptions should be defined on WHAT system abstracted views and HOW to manage them. The descriptions are strongly interlinked to system's abstractions. Multidisciplinary engineers in design practice commonly use black-box, grey-box and white-box terms but often disagree on the description details in these abstractions. The coupling matrix (CM) framework (represented in Figure 3 and illustrated in Figure 14 via ink-pen case study) is introduced to provide rigorous guidelines on WHAT and HOW by reasoning around system's descriptions (i.e. *goals to structure*) and abstraction views (i.e. *black-box to white-box*).

One of the key features of CM framework is that it couples two types of functions: environment-centric functions at black-box and device-centric (but solution agnostic) at grey-box and cascade them to solutions at white-box. Environment-centric functions of a system are defined in terms of its source and sink functions associated with external elements only. To substantiate this point, consider Figure 7 and Figure 8 from the ink-pen analysis in which system acts as source whose functions are define with only output flows (e.g. export ink). Also when system acts as a sink, its functions are represented with only input flows (e.g. import user effort and import ink). This concept of function is aligned with Multilevel Flow Modelling (Lind, 1994) and with the Interface Modelling Methodology (Uddin et al., 2016) and thus adopted for black-box analysis. Such concept of function is named as "**interaction functions**" by Uddin (2016) that exist between two interfacing elements where one element always being a system-of-interest for its designer. Device-centric functions are defined and conceptualised in two ways: (1) device's functions that will always have both inputs and outputs (see e.g. Stone and Wood, 2000) and (2) device's functions that would reside between internal elements of a device or system-of-interest (see e.g. Crilly, 2012). The former concept of device-centric function is adopted at grey-box where researchers argue about *solution agnostic* functions that can subsequently help in exploring many solutions. To substantiate this point, consider Figure 9 and 10 from the ink-pen analysis. This concept of function is argued at grey-box and is named as "**transformation functions**" by Uddin (2016). The other concept of device-centric function is aligned with Crilly (2012) where device-centric functions seem *solution specific*/driven and thus reside between internal elements. To substantiate this point, consider Figure 15 and 16. Such device-centric function concept is again referred as "interaction function" by Uddin (2016) that resides between two chosen interfacing elements. Therefore, device-centric functions can be further split into types (1) *solution agnostic* i.e. *transformation functions* and (2) *solution specific* i.e. *interaction functions* as shown in Figure 17. Figure 17 concludes the summary and scope of CM framework thereby describing which system's descriptions should be specified across which abstracted views in increasing order of detail.

The importance of black-box and white-box abstractions is now discussed in the context of behavioural descriptions. Brown and Blessing (2005) describe following behavioural constraints' description for the pen device; B1: ink flows from the tip; B2: ink coats the paper; B3: tip is moving. These behavioural descriptions represent flows coming from/to system or the internal elements behaviour that cause desired and undesired effects on the system's surroundings (e.g. paper). However, the aforementioned behavioural descriptions of B1 to B3 belong to different abstractions. To substantiate this, see for example Figure 8 in column S4 across interface D-E3, at black-box, designers can specify behaviour B2 which would result from pen and paper interface in the operation O5. Ink is the flow extracted from O5 which ultimately helps in defining environment-centric function as "Export ink from pen to paper". The desired effect on the paper is the ink flow.

The other behavioural descriptions B1 and B3 belong to pen's white-box abstraction where internal elements are known (such as nib/tip) and such behaviour descriptions (e.g. B1: ink flows from tip and B3: tip is moving) can only be specified at white-box and not at black-box where internals are not known yet. The undesired behaviour of a pen on paper such as hole creation can only be determined once internals are known. For example, an undesired behaviour "moving a nib/tip hard results in creation of

paper hole" and this can only be determined and specified at white-box that would yield to undesired effects as flows such as ink loss and paper surface loss. Such undesired behaviours descriptions are often solution or form dependent that can only be predicted at white-box. The undesired behaviours (e.g. ink loss and paper hole) in essence describe negative affordances of a chosen configuration of an ink-pen device. When designing, designers can only specify desired but intended behaviours mainly at black-box as the device is solution independent. Such kind of design reasoning between solution-agnostic (i.e. function-based) to solution/form-specific (affordance-based) concepts is discussed by Brown and Blessing (2005). Therefore, desired flows as intended behavioural effects' descriptions need to be described at black-box (Figure 5) of coupling framework and the undesired flows that can emerge based on chosen structural configuration can be specified only at white-box (Figure 15 and 16).

		Descriptions for a Multidisciplinary System				
		Goals	Actions	Functions	Behaviour	Structure
Abstractions for a Multidisciplinary System	Black-Box	<i>Goals</i>	<i>Interaction operation</i>	<i>Environment-centric but solution agnostic</i>		<i>Input and output flows only</i>
	Grey-Box				<i>Device-centric but solution agnostic</i>	<i>Input-output as well as flows' transitions</i>
	White-box				<i>Device centric but solution specific</i>	<i>Input-output as well as flows' transitions</i>

Figure 17. Descriptions mapping across abstractions of a multidisciplinary system

The CM framework has been tested across a number of multidisciplinary systems in an automotive company which has shown its promising strengths. A key strength has been that it couples multiple and necessary descriptions with relevant abstractions together thereby providing a cohesive overview of a multidisciplinary system. It aids the engineers in developing the understanding around the design ambition in increasing order of detail from black-box to white-box. The other key strength is that it enforces the designers to clearly define and differentiate between environment-centric and device-centric functions thereby establishing relationships in between them. The multidisciplinary systems (e.g. braking system, anti-roll system, propulsion system) are often acquired by automotive companies from various suppliers in their large projects (e.g. vehicles). For this purpose, the company's acquirers and the suppliers initially has to share a common understanding on environment-centric descriptions due to system being communicated at a black-box where its external surroundings and stakeholders are identified first. However, this view can represent partial functionality of the system. The discussion in between them then progresses to device-centric descriptions at grey-box that support in achieving the environment-centric descriptions needed by its stakeholders or external elements. The weakness of CM framework has been it often takes time to fill the matrices and to explore the coupling relations in between various descriptions which is also a common practice in engineering design. For example, the matrix-based approaches such as quality-function deployment (QFD) are often applied in engineering practice and frequently generate large matrices. It is no surprise that the same applies to CM framework, and hence, relatively large documents are expected while implementing the approach in many real-world complex systems such as automotive engines and aeroplanes.

5. Conclusions

This paper has introduced a structured CM framework for system analysis that aided the representation and integration of technical descriptions for a complex multidisciplinary system from goals to structure at its relevant abstractions from black-box to white-box. A key contribution of the paper is that it has established a rigorous framework and guidelines for multidisciplinary systems analysis built upon the gaps and baseline technical descriptions recommended in the existing frameworks and the five-layer conceptual model. In existing state of art, integration and traceability across several technical descriptions has been a challenging issue due to which matrix-based approaches has often been perceived to establish and trace relationships in between them as presented in Section 2. This paper also sets the strong academic foundations for CM framework by integrating the baseline descriptions available in existing matrix based approaches.

The working of CM approach is illustrated with a simple example of an ink-pen that has been repeatedly used by academic researchers to demonstrate their modelling concepts, and frameworks. The CM provides structured guidelines for representation of technical descriptions at relevant abstractions compared to the other matrix-based approaches. The CM framework can be regarded as an *effective* approach that explicitly models and integrates necessary technical descriptions at relevant key abstractions but at the same time its manual and document-driven nature makes it less *efficient*.

References

- Acharya, S. and Pandya, V. (2012), "Bridge between Black Box and White Box – Gray Box Testing Technique", *International Journal of Electronics and Computer Science Engineering*, Vol. 2 No. 1, pp. 175–185.
- Alexander, I. and Zink, T. (2002), "Introduction to Systems Engineering with Use Cases", *Computing and Control Engineering Journal*, pp. 289–297. <https://doi.org/10.1049/cce:20020607>
- Bonnema, G.M. (2008), FunKey Architecting, PhD thesis, University of Twente Enschede, The Netherlands.
- Brown, D.C. and Blessing, L. (2005), "The relationship between function and affordance", *Proceedings of ASME IDETC/CIE, California, USA*. <https://doi.org/10.1115/DETC2005-85017>
- Chandrasekaran, B. and Josephson, J.R. (2000), "Function in Device Representation", *Engineering with Computers*, Vol. 16 No. 3/4, pp. 162–177. <https://doi.org/10.1007/s003660070003>
- Crilly, N. (2012), "Function Propagation through Nested Systems", *Design Studies*, Vol. 34 No. 2, pp. 216–242. <https://doi.org/10.1016/j.destud.2012.10.003>
- Eisenbart, B. (2014), Supporting interdisciplinary system development through integrated function modelling, PhD thesis, University of Luxembourg.
- Hauser, J. and Clausing, D. (2009), *The House of Quality*, Harvard Business Review.
- Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, A. and Wood, K.L. (2002), "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts", *Research in Engineering Design*, Vol. 13 No. 2, pp. 65–82. <https://doi.org/10.1007/s00163-001-0008-3>
- King, A.M. and Sivaloganathan, S. (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*, Vol. 212 No. 3, pp. 215–230. <https://doi.org/10.1243/0954405981515635>
- Kossiakoff, A., Sweet, W.N., Seymour, S.J. and Biemer, S.M. (2011), *Systems Engineering Principles and Practice*, 2nd ed., John Wiley & Sons, New Jersey. <https://doi.org/10.1002/9781118001028>
- Lucero, B., Linsey, J. and Turner, C.J. (2016), "Frameworks for organising design performance metrics", *Journal of Engineering Design*, Vol. 27 No. 4-6, pp. 175-204. <https://doi.org/10.1080/09544828.2015.1135235>
- Otto, K. and Wood, K. (2001), *Product design: techniques in reverse engineering and new product development*, Prentice-Hall, New Jersey.
- Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Engineering Design: A Systematic Approach*, 3rd ed., Springer.
- Rational Software (2003), "*Rational Unified Process for Systems Engineering*", IBM Rational Software.
- Satzinger, J.W., Jackson, R.H. and Burd, S.D. (2012), *Systems analysis and design in a changing world*, 6th ed., Course Technology, Cengage Learning.
- Stone, R. and Wood, K. (2000), "Development of a functional basis for design", *Journal of Mechanical Design* Vol. 122 No. 4, pp. 359-370. <https://doi.org/10.1115/1.1289637>
- Suh, N.P. (2001), *Axiomatic Design: Advances and Applications*, Oxford University Press.
- Uddin, A. (2016), Development of an Integrated Interface modelling methodology to support system architecture analysis, PhD thesis, University of Bradford.
- Uddin, A., Campean, F. and Khan, M.K. (2016), "Application of the interface analysis template for deriving system requirements", *Proceedings of the DESIGN 2016 / 14th International Design Conference, Dubrovnik, Croatia, May 16–19, 2016*, pp. 543–552.
- Van Wie, M.J., Rajan, P., Campbell, M.I., Stone, R.B. and Wood, K.L. (2003), "Representing product architecture", *Proceedings of DETC03, Chicago, Illinois, USA*.
- Vermaas, P. (2009), "The Flexible Meaning of Function in Engineering", *International Conference on Engineering Design*, pp. 113–124.
- Yildirim, U. and Campean, F. (2014), "Development of a structured approach for decomposition of complex systems on a functional basis", *27th Int. Conf. on CAD/CAM-FoF*. <https://doi.org/10.1088/1757-899X/65/1/012004>

Dr. Amad Uddin, System Architecture Process Engineer
Jaguar Land Rover, Jaguar Land Rover
Banbury Road, Gaydon, CV35 0RR Warwick, United Kingdom
Email: uddinamad@live.com