Fall 2007

Enhanced functional analysis system technique for managing complex engineering projects

Sofia Tan

ENHANCED FUNCTIONAL ANALYSIS SYSTEM TECHNIQUE FOR MANAGING COMPLEX ENGINEERING PROJECTS

by

SOFIA TAN

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

2007

Approved by

Venkat Allada, Advisor

Cihan Dagli

Sahra Sedigh

PUBLICATION THESIS OPTION

This thesis will be submitted for publication to the INCOSE JOURNAL OF SYSTEMS ENGINEERING.

ABSTRACT

Functional analysis is an important aspect of the systems engineering process that provides the functional description of a system. Traditional functional analysis tools such as functional flow block diagrams (FFBD) progressively decompose functions into sub-functions based on considerations such as the operations sequence and customer requirements. However, as highlighted in the INCOSE Systems Engineering Handbook (2004), the FFBD does not provide full information relating to functional boxes and its interfaces, which are essential for the development of projects. The survey done by Pinto and Slevin (1990) suggested that one for the main reasons for projects failures is the lack of communication between the stakeholders.

This study presents an Enhanced Functional Analysis Systems Technique (EFAST) tool to facilitate communication amongst various stakeholders such as the customers, program managers, systems architects, and systems engineers. The EFAST maps the customer requirements to downstream system functions and subsystem/component requirements, and outlines the interactions between various system and subsystem level and activities using a top-down approach. A bottom-up approach is used to populate the system element cost and time estimates. The EFAST tool compares the budgeted development resources with the estimated development resources to provide a realistic picture for realizing project in terms of performance, cost, and schedule. The EFAST tool could potentially be used in the project bidding process because it compares the budgeted project cost with the estimated project cost. The application of the EFAST tool is demonstrated using a case example.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Allada, for giving me the opportunity to work with him, for his guidance, help, and constant encouragement, during the progressed of the research. I am especially thankful to Dr. Dagli for awarding me the graduate research funds to conduct this research and for his valuable thoughts to improve this research. Also, I would like to thank Dr. Sedigh for being a part of my committee and her suggestions to keep this research interesting. Their supervision and support was very much appreciated.

I would also like thank the Department of Engineering Management and Systems Engineering staff, particularly Wendy Berkelman for her assistance on coursework guidelines that contributes to this research. Also, many thanks to Mohit Goswami, who has help me tremendously in constructing the case study for this research, my roommate, Nandini, who has spent hours daily to prepare meals for us while I am finishing this research, and to all other SDL colleagues, Pradeep Tipaji, Abhijit Choudhury, Padma Parkala, and Harish Bagaitkar for their good humor and companionship as this research progressed.

Finally, I wish to thank my parents, especially my mum, who has given me her unconditional love, my brother and my sister who have provided me their great support. They have been very influential in all my efforts and accomplishments.

TABLE OF CONTENTS

Page

PUBLICATION THESIS OPTION ... iii

ABSTRACT ... iv

ACKNOWLEDGMENTS ... v

LIST OF ILLUSTRATIONS ... viii

LIST OF TABLES ... ix

NOMENCLATURE ... x

PAPER

ENHANCED FUNCTIONAL ANALYSIS SYSTEM TECHNIQUE FOR MANAGING COMPLEX ENGINEERING PROJECTS ... 1

ABSTRACT ... 1

1. INTRODUCTION ... 3

2. LITERATURE REVIEW ... 5

2.1. Functional Flow Block Diagram (FFBD) ... 5

2.2. Functional Analysis Systems Technique (FAST) ... 5

2.3. Cost Estimation ... 6

3. PROPOSED METHODOLOGY ... 7

3.1. Overview ... 8

3.2. Terms and Definitions ... 9

3.3. Phase 1: Functions and Components Decomposition ... 10

3.4. Phase 2: Concept Selection Analysis ... 11

3.5. Phase 3: Subassembly/Components Development Activities Planning and Scheduling ... 12

3.5.1. Design Information Flow Diagram (DIFD) ... 13

3.5.2. Work Breakdown Structure (WBS) ... 14

3.5.3. Activities Dependency Matrix (ADM) ... 15

3.5.4. Activities Sequence Diagram (ASD) ... 16

3.5.5. Network Diagram (ND) ... 17

3.6. Phase 4: Comparison of Budgeted Development Resources (BDR) with Estimated Development Resources (EDR) 18

4. CASE EXAMPLE 19

5. CONCLUSIONS AND FUTURE WORK 26

6. REFERENCES 27

VITA 51

LIST OF ILLUSTRATION

Figure | Page

1. Example of the functional flow block diagram 30
2. Proposed enhanced functional analysis systems technique methodology 31
3. Enhanced functional analysis systems technique system decomposition 32
4. Example of design information flow diagram (DIFD) 32
5. Example of activity dependency matrix (ADM) 33
6. Example of serial network 33
7. Example of parallel network 33
8. Example of network diagram (ND) 33
9. Application of EFAST on the truck case example 34
10. Ranking results from AHP analysis for truck case example 35
11. Application of DIFD on the truck case example 36
12. Application of WBS on the truck case example 37
13. Application of ADM on the truck case example 38
14. Application of ASD on the truck case example: Powertrain 39
15. Application of ND on the truck case example: Powertrain 40
16. Results for the worst case scenario 41
17. Results for the best case scenario 42
18. Results for the mid case scenario – cost overrun 43
19. Results for the mid case scenario – schedule overrun 43

LIST OF TABLES

Table Page

1. List of reasons and percentage of contribution to project failures 43
2. List of tools used by systems engineers and project managers 43
3. Engine design parameter ranges for the truck case example 44
4. Scale for AHP analysis 44
5. Pairwise comparison of design parameter on the truck case example 45
6. The relative ranking of design parameter on the truck case example 45
7. Pairwise comparison of the alternative concepts engine judged by RPM 46
8. The relative ranking of the alternative concepts engine judged by RPM 46
9. Pairwise comparison of the alternative concepts engine judged by torque 47
10. The relative ranking of the alternative concepts engine judged by torque 47
11. Pairwise comparison of the alternative concepts engine judged by HP 48
12. The relative ranking of the alternative concepts engine judged by HP 48
13. Estimated cost (EC) and budgeted cost (BC) 49
14. Estimated time (ET) and budgeted time (BT) 50

NOMENCLATURE

Symbol	Description
DT	Delay time between two activities in days
d_i	Duration time to complete development activity i in days
ES_{i+1}	Early start time for development activity $i+1$
EF_i	Early finish time for development activity i
LS_i	Late start time for development activity i
LF_i	Late finish time for development activity i

ENHANCED FUNCTIONAL ANALYSIS SYSTEM TECHNIQUE FOR MANAGING COMPLEX ENGINEERING PROJECTS

Sofia Tan, Venkat Allada
University of Missouri – Rolla, Missouri, U.S.A 65401
Email: st3g7@umr.edu, allada@umr.edu

ABSTRACT

Functional analysis is an important aspect of the systems engineering process that provides the functional description of a system. Traditional functional analysis tools such as the functional flow block diagrams (FFBD) progressively decompose functions into sub-functions based on considerations such as the operations sequence and customer requirements. However, as highlighted in the INCOSE Systems Engineering Handbook (2004), the FFBD does not provide full information relating to functional boxes and its interfaces, which are essential for the development of projects. A survey done by Pinto and Slevin (1990) suggested that one for the main reasons for projects failures is the lack of communication between the stakeholders.

This study presents an Enhanced Functional Analysis Systems Technique (EFAST) to facilitate communication amongst various stakeholders such as the customers, program managers, systems architects, and systems engineers. The EFAST maps the customer requirements to downstream system functions and subsystem/component requirements, and outlines the interactions between various system and subsystem level and activities using a top-down approach. A bottom-up approach is used to populate the system element cost and time estimates. The EFAST tool compares the budgeted development resources with the estimated development resources to provide a realistic picture for realizing projects in terms of performance, cost, and schedule. The EFAST tool can potentially be used in the project bidding process because it compares the budgeted project cost with the estimated project cost. The application of the EFAST tool is demonstrated using a case example.

Keywords

Functional Analysis, Engineering Projects, Systems Engineering.

1. INTRODUCTION

The design of complex system architectures involve a significant amount of resources commitments. Typically, the systems architects are responsible for determining the functional requirements by compiling the unstructured mix of customers' needs, ideas, requests, and technological possibilities into a coherent and structured system. As mentioned in the INCOSE Systems Engineering Handbook (2004), functional analysis should be conducted to define and integrate a functional architecture, for which subsystems and activities can be assigned. Furthermore, the analysis must be conducted to a particular level of depth, which is needed to support the design synthesis efforts. Hence, it is essential for systems architects to perform a high-level quantitative analysis to determine system feasibility.

During the system integration phase, systems engineers are responsible for finding the optimal design solutions. This is done by breaking the system level requirements into subsystem level requirements to identify input design criterions and/or constraints on various elements of the system. Consequently, the systems architects must fully determine the overall system objectives governed by the customers and transfer this information to systems engineers to analyze the subsystem requirements and constraints. The system project managers have the task of managing the entire project and evaluating that the project cost, time, and performance objectives of the customer are met. However, this is often the most difficult phases because it requires a systems thinking approach to gather important information effectively for systems engineers, systems architects, and project managers. Due to different information priorities amongst systems engineers, systems architects, and project managers, often times, these internal stakeholders analyze the system from different perspectives. Systems architects and engineers frequently look at accomplishing the activities for a system by breaking down the system level requirements into subsystem level requirements in a hierarchical structure. The project mangers look at the overall project development activities to ensure that feasible resources and budgets are properly identified for each activity to meet the customer needs. When the overall project budgets and timeline fail to meet the customer needs, it is highly probable that the project overruns, thereby, causing a project failure.

Refer to Table I. Table I provides a list of prominent reasons for project failures. The major cause of projects failure is attributed to incomplete requirements and lack of user involvement. The incomplete requirements and lack of user involvement in projects are often due to lack of communication. Therefore, an effective representation tool is needed for clear communication between the systems engineers, architects, and the project managers. This study proposes one such tool called the EFAST to facilitate communication between systems architects, systems designers/engineers, and project managers. Refer to Table II. Table II summarizes commonly used tools by system architects/engineers and project managers.

2. LITERATURE REVIEW

2.1 Functional Flow Block Diagram (FFBD)

The Functional Flow Block Diagram (FFBD) uses a multi-tier and step-by-step diagram of the system functional flow to define the detailed operational sequences of the system functions (INCOSE Systems Engineering Handbook, 2004). It is a commonly used tool in functional analysis to define the functional level and the sequences of activities. The decomposition of the function into the sequence of activities is carried out by asking the question "WHAT" needs to be done to perform the particular function. Refer to Figure 1. The top level functions of the system are shown at level 1. At level 2, the top level function, F1, is decomposed into level 2 functions, F1.1 through F1.n. The functional decomposition continues to further levels as dictated by the scope of the study.

One of the limitations of FFBD is that it does not provide the information for each functional step and the timeline details. As the system development progresses, the functional requirements will change to accommodate the resource constraints, hence, the FFBD has to be updated frequently to ensure that the latest system architecture is depicted.

2.2 Functional Analysis Systems Technique (FAST)

Functional Analysis Systems Technique (FAST) is a requirement oriented and functional based tool, which focuses on the functions required by a design, process, or service to accomplish its objective (Wixson, 1999). FAST is one of the synergistic ways of developing, decomposing, and understanding the system functions, hence, it can play an important role within the context of systems engineering.

The FAST modeling process starts by identifying the system's primary objective and basic function(s). The basic function(s) are decomposed into secondary support functions, and finally, into the supporting functions to support the basic functions. The secondary functions are the ones required for supporting the primary functions. The FAST diagram answers the question "HOW" while moving from the left to the right and answers the question "WHY" while moving from the right to the left to ensure a logical formation of functional relationships.

2.3 Cost Estimation

Cost estimation is one of the most crucial and difficult process in a system development. Without accurate cost estimation, the project is at a risk of overruns. Studies done by Standish Group and Scientific American from 1994 through 1996, which evaluated about 300 complex projects, suggested that approximately 53% of the complex projects overrun by approximately 89% of their original cost. The study also mentioned that average time overrun was approximately 122% of their original schedule. Therefore, it is important to plan and control the project activities right from the beginning.

Often times, the customer demands constrain the project development, and it is the responsibility of the systems designers/engineers to work with the constraints imposed. Consequently, a good cost estimation model is required to assist the system designers/engineers to calculate the most feasible resources required to the most desired system capability and performance. Cost estimation can also assist the system designers/engineers with further analysis such as tradeoff studies and risk analysis.

Over the years, many techniques have been introduced to assist software designers in cost estimation for software development. The methods available for estimating cost include algorithmic techniques, analogy estimating techniques, expert judgment methods, bottom-up and top-down approaches (Wu, 1997). In the survey done by Chulani (1998), the most commonly used software cost estimating models are the Putnam model, COCOMO model, and function points based model. System development is similar to software development and there is no difference in estimating cost for system development (Wu, 1997). Hence, software cost estimation methods can be adapted to component cost estimation.

3. PROPOSED METHODOLOGY

3.1. Overview

The foundation of the proposed EFAST model is based on the Functional Analysis Systems Technique (FAST) representation tool that is widely used by value analysts. The proposed EFAST tool uses a top-down approach and a bottom-up approach for allocation of system development resources. The development resources are project cost and time. This model will allow the managers and engineers to continuously predict the future needs of the evolving system and technical intricacies as well as allocation of future development resources.

The top-down approach allocates the system development resources to various elements of the system starting from high level system to lower level subsystems. In EFAST, the allocation of development resources using the top-down approach are the total budgeted cost (BC) and total budgeted time (BT). The EFAST tool assumes that the BC and BT is provided by the customer, and is allocated to each of the identified functions and sub-functions.

In the bottom-up approach, the costs for each subassembly/component development activities are estimated and all these costs are then aggregated to provide estimated cost of the overall system. The bottom-up approach uses historical data from similar engineering projects to estimate the costs, revenues, and other data for the current project by using appropriate modification factors (Sullivan, *et al.*, 2005). William (1994) stressed the importance of establishing the work breakdown structure (WBS) before the bottom-up approach is applied. Hence, a WBS technique is used to define the subassembly component development activities, estimated cost (EC) and estimated time (ET) for each development activity. Figure 2 outlines the overall EFAST steps.

In the first phase of EFAST, a FAST diagram is developed. The focus is on finding the system requirements that fulfill the customer needs. The subsystem structure is identified, which is further broken down into component level structures. The second phase consists of finding the alternative components for each subassembly/component.

The concept selection analysis is conducted on the alternative components to identify the best alternative that meets the customer needs. It is assumed that the customer needs are already defined and ranked based on stakeholders importance before conducting the concept selection analysis. In the third phase, the selected alternative component is broken down into subassembly/component development activities.

The development activities will be planned and scheduled by the engineers, and the estimated development resources (EDR) are allocated to each development activities. The estimated development resources (EDR) will include estimated cost (EC) and estimated time (ET). The fourth phase is comparing the estimated development resources (EDR) with the budgeted development resources (BDR). Finally, the results are evaluated to determine the feasibility of the project meeting the customer objectives.

3.2. Terms and Definitions

Top-down approach: Top-down approach is a strategy that looks at the entire system concept and breakdowns the system into subsystems.

Bottom-up approach: Bottom-up approach is a strategy that defines functional details of the smallest element beforehand in a particular system and further links it to higher-level elements, and finally a larger system is formed.

Subassembly component development activities: Subassembly component development activities are a set of activity that must be executed when structuring a subassembly component.

Alternative subassembly/components: Alternative subassembly/components are the choices available for the subassembly/components.

Concept selection: Concept selection is an act to select the best concept to perform the function requirements from a set of alternatives.

Estimated development resources (EDR): Estimated development resources are the calculated development resources required to develop a particular system. It is calculated based on the past experiences of the technical experts and usually includes funds, personnel, information, etc.

Budgeted development resources (BDR): Budgeted development resources are the planned development resources for a particular project. It is based on the customer specified requirements such as expenditures and delivery schedule, and it is allocated by the management to the entire project.

3.3. Phase 1: Functions and Components Decomposition

Step 1: Define the objective and the primary functions.

Step 2: Define secondary functions and supporting secondary functions.

Step 3: Identify subsystems structure and subassembly/components for the terminal functional boxes.

The first phase of the EFAST model is to conduct functional decomposition. In this step, the FAST method is used to decompose into functions by finding the inputs and outputs that are required to achieve the overall system requirements. Refer to Figure 3. Figure 3 provides an example of function decomposition hierarchy.

Apart from functional decomposition which is provided by the traditional FAST diagram, the proposed EFAST tool can be used for the following tasks:

- Function and structure data boxes that list information such as function cost and function completion time.
- Extension of the terminal function block in the traditional FAST diagram to manifest potential alternative physical structure solutions.
- Selection of the best component based on multi-objective criteria.

3.4. Phase 2: Concept Selection Analysis

Step 4: Conduct concept selection analysis on alternative structural concepts.

Concept selection is an important aspect of the decision-making process and is used to evaluate alternative concept solution based on customer needs to assess the feasibility in realizing the design. This process involves comparing the relative strengths and weaknesses of the alternative concepts and the selected concept is used for further development (Adrian, *et al*., 2007). There are many methods to assist the system designers to obtain the best results in the concept selection phase. The commonly used methods are listed below:

1. Quality Function Deployment (QFD)
2. Pugh's Scoring Analysis
3. Axiomatic Design
4. Analytical Hierarchy Process (AHP)

In EFAST, the concept selection analysis is conducted on the identified alternative subassembly/components. The selected alternative subassembly/component is then further analyzed in steps 5 through 8, where the development activities interaction will be identified and broken down to facilitate the allocation of estimated development cost and time. Step 4 extends the terminal functional blocks to manifest various structural concepts. For illustration purposes, the AHP is employed in the case example which will be discussed in the later section.

3.5. Phase 3: Subassembly/Components Development Activities Planning and Scheduling

Step 5: Construct the Design Information Flow Diagram (DIFD) to illustrate interaction and information dependency between the subassembly/components.

Step 6: Construct the Work Breakdown Structure (WBS) for each subassembly/components.

Step 7: Construct the Activities Dependency Matrix (ADM) to illustrate the dependency between subassembly/components development activities.

Step 8: Construct the Activities Sequence Diagram (ASD) to allocate cost and time for each development activity and construct the network diagram to provide timing details.

3.5.1. Design Information Flow Diagram (DIFD)

After the identification of all the best subassembly/components, Design Information Flow Diagram (DIFD) is constructed to illustrate the interaction and information dependency between the subassembly/components. DIFD lists the subsystems, subassembly/components and the point of information exchange. The example of DIFD is shown in Figure 4. The point of information exchange denotes the percentage of development activities of primary subassembly component (C_p) must be completed in order to transfer the design information to the dependent subassembly/components (C_d). In the example, the value *x* is the percentage of development activities must be completed. The direction of the arrow represents the direction of the information flowing from C_p to C_d. The steps for constructing the DIFD are listed as follows:

1. List the subassembly/components on the X-axis, and the percentage of development activities completed on the Y-axis.
2. Identify the primary and dependent subassembly/components.
3. Approximate the percentage development activities of C_p that must complete.
4. Assign the point of design interaction from C_p to C_d.

DIFD is used to find the delay period between the development activities of components, which are governed by the finish-to-start relationships. The finish-to-start relationship refers that the development of C_d cannot start until the development of C_p is completed.

In case of any design changes in a particular subassembly component, DIFD is also capable of illustrating the impact of the design changes on the dependent subassembly/components. For example, if a certain design specification for primary component, C_p is needs to be changed, the DIFD allows a quick reference to identify all components, which are dependent on that component.

3.5.2 Work Breakdown Structure (WBS)

Developing a complex system involves breaking down the set of development activities required for completion of the project. WBS is developed using top-down approach in successive levels of detail (Sullivan, *et al.,* 2005). The first step structuring WBS involves breaking down the system into its major subsystems (Level 2), and then will be further decomposed into subassembly/components (Level 3) and so on. For example, in the truck development project, the truck system is divided into second-level subsystems such as the powertrain, load bearing units, body and auxiliary units. Each second-level subsystem of the WBS can be further subdivided into the third level. For example, the powertrain can be subdivided into third-level components such as engine, gearbox, propeller shaft, fuel tank, and clutch. This process continues until the details of the subassembly component development activities of the system are accomplished.

During this process, the numbering scheme is used to indicate the interrelationships of the development activities in the hierarchy and to facilitate the manipulation and integration of data (Sullivan, *et al.*, 2005).

However, the WBS does not provide the timeline of the development activities. Hence, the network diagram (ND) is used to illustrate the timeline required for system development. In EFAST, the network diagram (ND) will be constructed after the sequence of development activities is identified.

3.5.3 Activities Dependency Matrix (ADM)

Activities Dependency Matrix (ADM) provides information exchange pertaining to development activities, such as percentage development activities completed, and its information dependency. In the matrix, the rows and the columns represent the development activities. The fraction delay time (DT) of the interacting development activities is located in the right side of the diagonal cells. For example, to develop component C_p, two development activities, namely, ACT1 and ACT2 need to be completed. Further, Z_{act1}% of activity ACT1 must be completed for transferring information in order to start ACT2. Figure 5 illustrates the construction of ADM. The delay time (DT) to start activity ACT2 is calculated using equation (1):

$$DT = Z_{act1}\,\% \times d \text{ of ACT 1} \qquad \text{Eq.(1)}$$

3.5.4. Activities Sequence Diagram (ASD)

The sequence of the development activities for each subassembly component is identified using the ASD. The sequence information obtained from the ASD will be used to assist the structuring of the ND. The development activities can be predecessor activities, which must be completed prior to the start of the particular activity, or successor activities, which cannot start until a particular activity is completed.

After the development activities of each subassembly component are identified, the estimated cost (EC) and estimated time (ET) are allocated to each activity in the activities sequence diagram (ASD). For illustration purposes, the estimating by analogy is applied to the case example. Estimating by analogy estimates the current project costs by comparing it with previous similar project. This method of cost estimation is usually based on the estimator's past experience and the historical data of previous project.

Later, the network diagram (ND) is drawn to show the timeline to develop the overall system. Stephen (2002) defined a serial and a parallel network as follows:

Series Network: Two activities are in serial when one is a predecessor of the other. The boxes will be used to represent the development activities. Figure 6 shows the detail of a typical serial network.

Parallel Network: Two activities are in parallel, if neither is a predecessor or a successor of the other. Figure 7 shows details of a typical parallel network.

3.5.5. Network Diagram (ND)

When constructing a ND, the duration time (d), early start (ES) and early finish (EF), and late start (LS) and late finish (LF) are identified. The ES and EF times for each development activity are calculated by moving forward through the network and determining the earliest time at which an activity can start and finish considering its predecessor activities. The LS and LF times indicates the latest time an activity can start and finish without delaying the total time completion of the project. LS and LF are calculated by moving back through the network.

The difference between the late and early finish of each activity is the activity's delay. The critical path is the path through the network in which none of the activities have delays. The total project completion time can be calculated by summing the completion times of the activities in the critical path (Howard, 2004). By summing the data, the probability of the project completed according to the planned schedule can be identified. The equation for the calculations of ES, EF, LS, and LF is as follows:

$$ES_{i+1} = (d_i \times DT\%) + ES_i \qquad \text{Eq.(2)}$$

$$EF_{i+1} = d_{i+1} + ES_{i+1} \qquad \text{Eq.(3)}$$

$$LS_i = (- d_i \times DT\%_{max}) + LS_{i+1} \qquad \text{Eq.(4)}$$

$$LF_i = LS_i + d_i \qquad \text{Eq.(5)}$$

Figure 8 shows an example of a network diagram. After completion of the network diagram, the delay time can be identified. The proceeding step involves comparing the budgeted development resources with the estimated development resources.

3.6. Phase 4: Comparison of Budgeted Development Resources (BDR) with Estimated Development Resources (EDR)

Phase 1 provides the top-down approach for allocating the budgeted development resource (BDR), and phase 3 provides the bottom-up approach to calculate the estimated development resources (EDR). In phase 4, the results obtained from both the top down and the bottom up approach are compared.

There are four possible case scenarios that could occur in a project development. The four case scenarios are listed as follows:

1. Worst Case Scenario: The EDR does not meet the BDR or project overruns
2. Best Case Scenario: The EDR meets the BDR or project success
3. Mid Case Scenario (cost): EC does not meet the BC or project cost overruns
4. Mid Case Scenario (schedule): ET does not meet the BT or project time overruns

The results from the comparison and the generated case scenarios could trigger further analysis such as cost risk analysis, schedule risk analysis, performance risk analysis, PERT/CPM, and so forth for the projects. In the next section, the application of EFAST to a truck system project case is demonstrated.

4. CASE EXAMPLE

The EFAST tool is demonstrated using a midsize truck system. Refer to Figure 10. EFAST modeling starts by identifying the five types of boxes that are used in the EFAST diagram, namely, customer needs, function box, terminal function box, subsystem structure box, and subassembly component box.

PHASE 1

Step 1: Define the objective and the primary functions.

The identification of customer needs starts by asking a few questions such as the following:

a. What is the main objective of the project?

b. What are the high-level solutions necessary to perform this objective?

The general solutions identified are the high-level functions or the primary functions as shown in Figure 9. The objective for the development of commercial truck is identified as "Develop Truck System", while the primary functions are as follows*: move vehicle*, *support vehicle load, support load and driver,* and *maneuver vehicle and stop vehicle*.

Step 2: Define secondary functions and supporting secondary functions.

Refer to Figure 9. For the primary function, *move vehicle*, the secondary function has been identified: *generate power*. The supporting secondary functions for *generate power* is *convert energy*. Depending on the case, the supporting secondary functions can be decomposed into several levels. Referring to Figure 10, the supporting secondary function *convert energy* is called the terminal functional box because no further function decomposition occurs after this point.

Step 3: Identify subsystems structure and subassembly/components for the terminal functions boxes.

Refer to Figure 9. Corresponding to the terminal function box *convert energy*, the identified subsystem structure that domain the task is powertrain. For this subsystem structure, five subassembly/components have been identified namely: engine, gear box, propeller shaft, fuel tank, and clutch.

In this step, the functional data box for supporting secondary functions *convert energy* is created which includes the following information:

- Budgeted Cost (BC)
- Budgeted Time (BT)

The budgeted function cost and schedule is allocated to the functional data box by systems architect. For this case, the BC and BT is based on historical data. The BC allocated is $80.7M, and the BT allocated is 570 days.

PHASE 2

Step 4: Conduct concept selection analysis on alternatives subassembly/component.

Refer to Figure 9. The subassembly/component engine is further analyzed. Analytical Hierarchy Process (AHP) is used to demonstrate the concept selection analysis. In this case example, it is assumed that the customer criteria for the subassembly/component engine are fuel economy, power-to-weight ratio, and noise vibration and harshness. The customer criteria will then be evaluated based on the design parameters such as horsepower, rotation-per-minute, and torque. Refer to Table III. Table III lists the design parameters requirement ranges that satisfy the customer criteria. Refer to Table IV. The design parameters are scored according to the customer criteria using the scale listed in Table IV.

Table V through Table XII illustrates the pairwise comparison of the alternative concepts with each design parameters. Figure 10 provides the results of the overall analysis. From this analysis, it is evident that the turbocharged inter cooled engine is preferred over the other two existing alternatives i.e., naturally aspirated engine, and turbocharged engine.

PHASE 3

Step 5: Construct the Design Information Flow Diagram.

Refer to Figure 11. The development activities of subassembly/component engine are broken down into percentage of development activities that are completed. Figure 11 shows that after 10% of engine development activities are completed, the information is transferred from engine to fuel tank, wheels and tires, and steering. The point of information flow will be used in the later steps to find the delay time between the subassembly/component development activities.

Step 6: Work Breakdown Structure (WBS) of development activities of each subsystem's component.

Refer to Figure 12. The truck system is broken down into its subsystem structures (WBS Level 2) namely, powertrain, load bearing, body and auxiliary. Then, each subsystem structure is further divided into subassembly component (WBS Level 3). For example, the powertrain is subdivided into third-level components namely the engine, gearbox, propeller shaft, fuel tank, and clutch. Finally, the components are mapped into development activities of each subassembly/component.

Step 7: Develop Activities Dependency Matrix.

Refer to Figure 13. The Activities Dependency Matrix (ADM) for the truck system is constructed. In the large matrix, the rows and the columns represent the subassembly component. In the smaller matrix, it represents the development activities for each subassembly component. The delay time (DT) of the interacting development activities is located on the right side of the diagonal cells. For example, to develop the subassembly/component engine, six development activities namely 1.1.1.1 through 1.1.1.6 need to be completed. Further, 20% of the development activity 1.1.1.1 needs to be completed for transferring the information in order to start the development activity 1.1.1.2. Hence, the total delay to start the activity 1.1.1.2 can be calculated using eq.(1):

$DT_{1.1.1.2} = 0.2 \times d_{1.1.1.1} = 0.2 \times 600 = 120$ days

The total development time is equivalent to the ET, which is estimated by the engineers. Similarly, the step is applied to the overall truck system, and the information obtained is used in the Network Diagram.

Step 8: Develop the Activities Sequence Diagram (ASD) to allocate cost and time for each breakdown development activities and Network Diagram.

Refer to Figure 14. The subassembly/component turbocharged inter-cooled engine is used for further analysis. The development activities of the turbocharged inter-cooled engine are broken down, and then EC and ET are allocated to each of the development activities. Similarly, these steps are applied to the other subassembly/components. The ASD provides information of development activities sequences of each subassembly/component to the design engineers, and allows the design engineers to allocate the resources according to the development activities sequences. Then, the structure data box for powertrain subsystem is created to include the following information:

- Estimated Cost (EC)
- Estimated Time (ET)

For illustration purposes, estimating by analogy is applied in this step. The total estimated structure cost for powertrain subsystem is based on the total estimated cost of the development activities for the respective subsystem. Similarly, the total estimated time to complete the powertrain subsystem is derived from summing the estimated completion time of development activities for the powertrain subsystem.

Refer to Figure 15. The network diagram for subsystem powertrain is constructed. The first development activity for subsystem powertrain is activity 1.1.1.1 (engine core). The duration time, (d) to develop the activity 1.1.1.1 is identified as 600 days. Since it is the first activity to start, the early start (ES) will be zero. However, the earliest finish can be calculated using eq.(3):

$EF_{1.1.1.1} = d_{1.1.1.1} + ES_{1.1.1.1} \rightarrow 600 + 0 = 600$ days

Moving forward through the network, the next activity is 1.1.1.2. The ES and EF for activity 1.1.1.2 can be calculated as follows:

$ES_{1.1.1.2} = (d_{1.1.1.1} \text{ x } DT\%) + ES_{1.1.1.1} \rightarrow 600(0.2) + 0 = 120$ days

$EF_{1.1.1.2} = \mathrm{d}_{1.1.1.2} + ES_{1.1.1.2} \rightarrow 320 + 120 = 440$ days

Likewise, the same method is applied for the rest of the network diagram. The LS and LF can be calculated by moving back through the network. Using the activity 1.1.1.6 as the last development activity, the LF is the equivalent to its EF, which equals 630 days. The LS can be calculated using eq.(4):

$LS_{1.1.1.6} = (- d_{1.1.1.6} \text{ x } DT\ \%_{max}) + LS_{\ i+1}$ → (- 50 x 1 + 630) = 580 days

However, for the case of activity 1.1.2.1, the LS and LF needed to be calculated using eq.(4) and eq.(5). The LS and LF calculation is shown below:

$LS_{1.1.2.1} = (- d_{1.1.2.1} \text{ x } DT\%_{\ max}) + LS_{\ 1.1.3.1}$ → - 98(0.4) + 510 = 470 days

$LF_{1.1.2.1} = LS_{1.1.2.1} + d_{\ 1.1.2.1}$ → 470 + 98 = 568 days

The same method is applied to rest of the network diagram. The critical path is the path through the network in which none of the activities have delays. In the case of subsystem engine development, the critical path is identified as shown below:

1.1.1.1 → 1.1.1.4 → 1.1.1.5 → 1.1.1.6

The total project completion time can be calculated by summing the variances in the completion times of the activities in the critical path. In this case, the total project completion time is 630 days.

Step 9: Comparison of budgeted development resources and estimated development resources.

Refer to Table XIII and Table XIV. The tables show the comparison of estimated development resources (EDR) with budgeted development resources (BDR). For example, in the powertrain subsystem, the estimated cost (EC) is calculated as $82.4M and the budgeted cost (BC) given as $80.7M. This comparison shows that there is a difference of $1.7M in monetary resources. Similarly, a comparison can be done with other subsystems.

Refer to Figure 16 through Figure 19, which compares the EDR and BDR in graphs. The graph shows four possible case scenarios that could occur in projects. Scenarios 1, 3, and 4 fall in the category of high-risk, where the estimated development resources does not meet the customer requirements. The results from the graphs could potentially set off further analysis such as risk analysis, PERT/CPM, and so forth.

5. CONCLUSIONS AND FUTURE WORK

This study presents the initial framework of the EFAST tool that can serve as a viable representation and communication tool for the system architects, systems engineers, and program managers. The proposed EFAST tool can provide an efficient communication forum between multiple stakeholders because it describes information from two different perspectives, namely, engineering viewpoint (bottom-up approach) and project management viewpoint (top-down approach). The EFAST representation provides information for the project managers regarding the feasibility of the system development in terms of cost and schedule.

The EFAST tool can include other analysis such as schedule risk analysis, cost risk analysis, and performance risk analysis. It can also be extended to indicate the alternative functional decompositions. The EFAST tool could be modified to create more comprehensive functional and structural data boxes capable of storing richer attributes, and can be extended to provide requirement traceability along various functional and structural box routes. This will help the system designers to clearly communicate with the project managers on the implications of changes in requirements on the system level performance and project management metrics. Additionally, the EFAST can include a comprehensive approach to estimate the subassembly component development activities cost using software estimation cost techniques.

6. REFERENCES

S. Michael, Quantifying Uncertainty in Multi Criteria Concept Selection Methods, Journal Engineering Design 17 (4) (2007), 175 – 187.

B James, Systems Engineering – A Retrospective View, J Sys Eng 1(4), 258 – 266.

Systems Engineering Handbook, version 2a, Technical Board International Council of Systems Engineering (INCOSE), Seattle, WA, 2004.

U. Karl and E. Steven, Product Design and Development, McGraw-Hill, New York, 2003, 134 – 137.

W. James, Functional Analysis and Decomposition using Functional Analysis Systems Technique, Proceedings International Council on Systems Engineering Annual Conference, Idaho, INEEL/COM-98-01072 (1999), 1- 6.

B. Ben and F. Walter, Systems Engineering and Analysis, Prentice Hall, New York, 2006, 173 – 179.

M. James, Systems Engineering Guidebook: A Process for Developing Systems and Products, CRC Press, Boca Raton, 2000, 71 – 85.

F. Ian and R.Michael, Managing Complex Technical Projects: A Systems Engineering Approach, Artech House, New York, 2003, 51 – 54.

S. Larry and M. David, Value: Its Measurement, Design and Management, John Wiley & Son, New York, 1992.

P. Michael, Scheduling Theory, Algorithms, and Systems, Prentice Hall, New Jersey, 2002, 398 – 406.

T.R Browning, Modeling and analyzing cost, schedule, and performance in complex system product development, Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA, 1998.

T.R Browning, Applying the Design Structure Matrix to System Decomposition and Integration Problems: A review and new directions, IEEE Transactions on Engineering Management 48(3) (2001), 292 – 306.

P.K Jeffrey and M.J Samuel, The Causes of Project Failure, IEEE Transactions on Engineering Management 37(4) (1990), 269 – 276.

E. Howard, Essential of Project and Systems Engineering Management 2nd Edition, Wiley & Sons Inc. New York, 2005, 67 – 91.

Y. Tsuneo, K. Thoru, A Framework for Top-down Cost Estimation of Software Development, IEEE Transactions on Software Engineering 9(11) (1999), 322 – 323.

W.G William, Development Cost Prediction, IEE Colloquium on Life Cycle Costing and Business Plan, INSPEC accession ref: 4653319 (1994), 411 - 414.

B. Barry *et al*., Software Development Cost Estimation Approaches: A Survey, IBM Research, California, 1998, 1 – 46.

T.R Browning and E.D Steven, Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development, IEEE Transactions on Engineering Management 49(4) (2002), 428 – 442.

E.D Steven and S.Vesa, Patterns of Product Development Interactions, Int. Conf. on Eng. Design, Glasgow (2001), http://hdl.handle.net/1721.1/3808, last accessed: August 8, 2007.

T.R Browning *et al.*, Adding Value by Creating Information and Reducing Risk, IEEE Trans on Eng Management 49(4) (2002), 443 – 458.

E.L. Cole, Functional Analysis: A System Conceptual Design Tool, IEEE Transactions on Aerospace and Electronic System 34(2) (1998), 354 – 365.

X. Adrian, *et al.*, A Comparison of Concept Selection in Concept Scoring & Axiomatic Design (2007), http://cden2007.eng.umanitoba.ca/resources/papers/69.pdf, last accessed: August 20, 2007.

W. Liming, The Comparison of the Software Cost Estimating Methods (1997), http://www.computing.dcu.ie/~renaat/ca421/LWu1.html, last accessed: August 19, 2007.

E.D, Steven, A Planning Method for Integration of Large-Scale Engineering Systems, International Conference on Engineering Design Vol. 25 (1997), 199 -204.

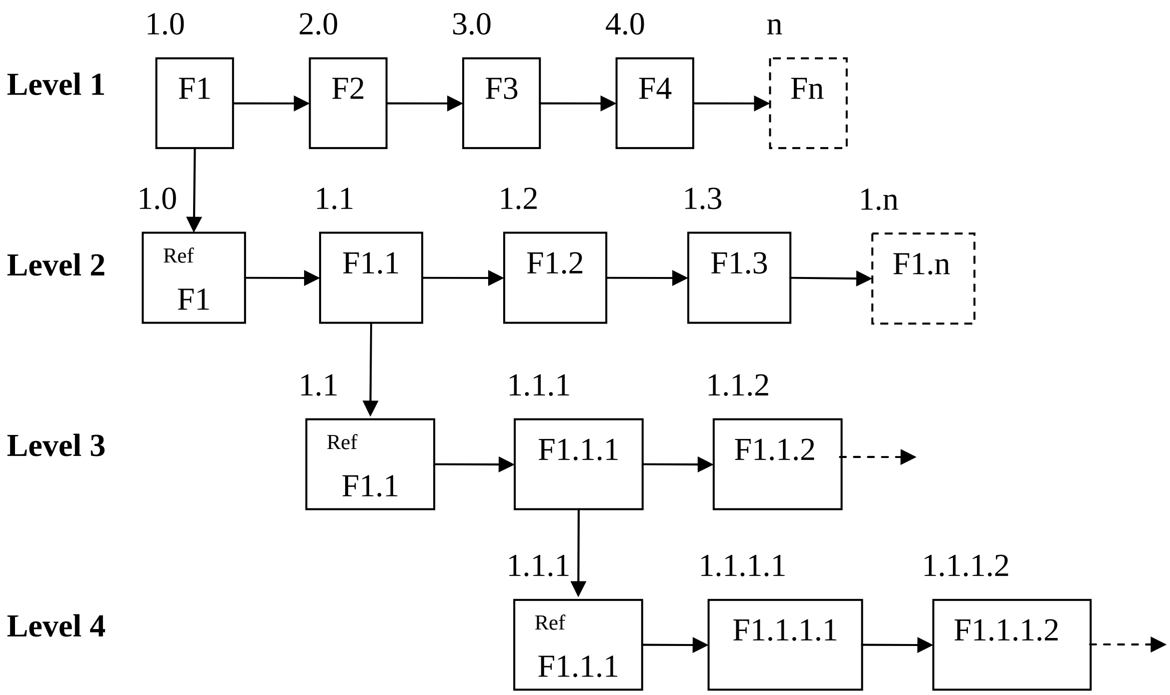

Figure 1: Example of the functional flow block diagram

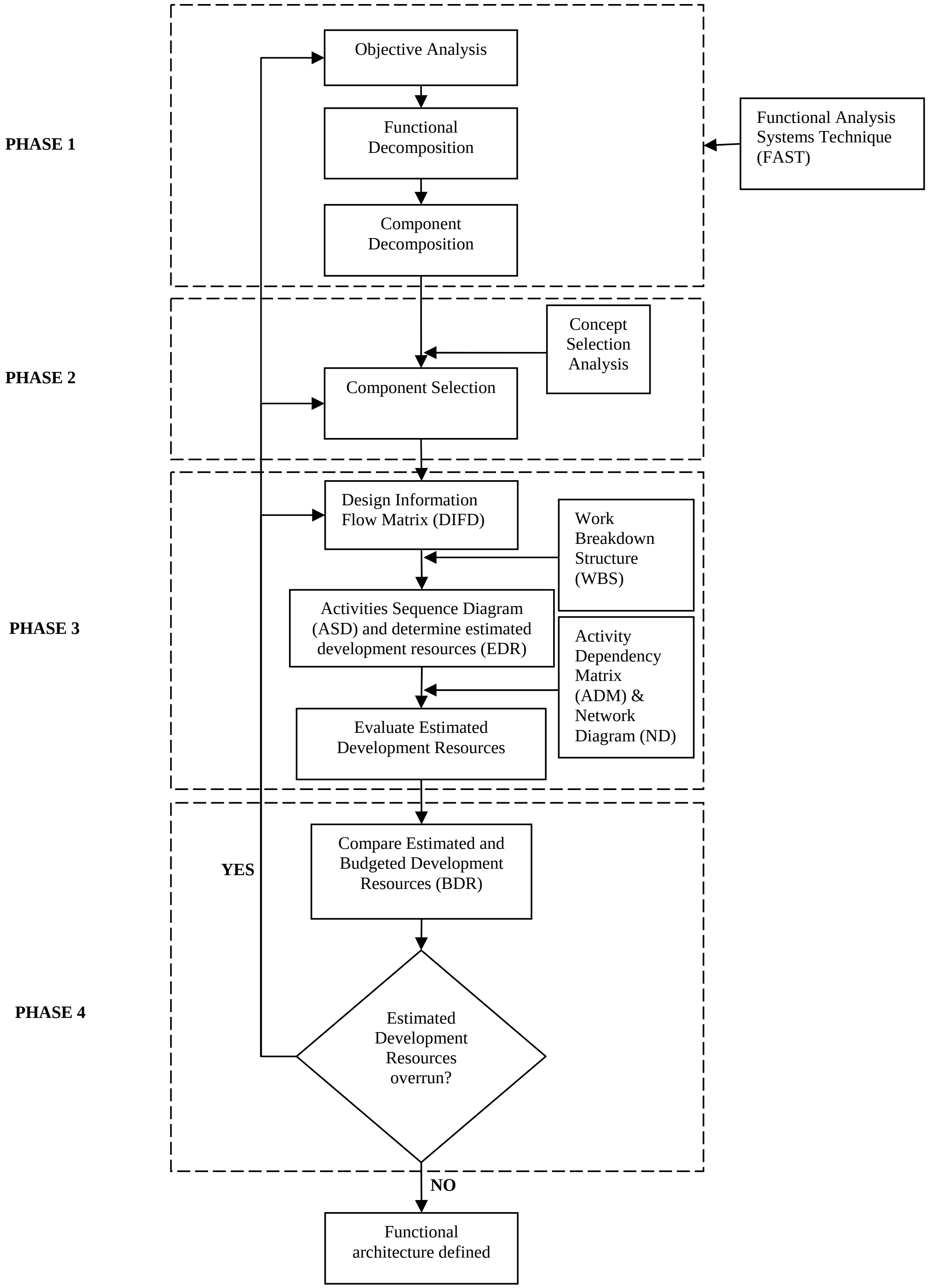

Figure 2. Proposed enhanced functional analysis systems technique methodology

Primary function: Highest level of system objectives/customer objectives.

Secondary function: Decomposed function that support system objective.

Supporting secondary function: Decomposed function that support secondary function.

Subsystem structure: Identified subsystems to perform the supporting secondary function.

Subassembly/Component: Components that are needed to support the subsystem.

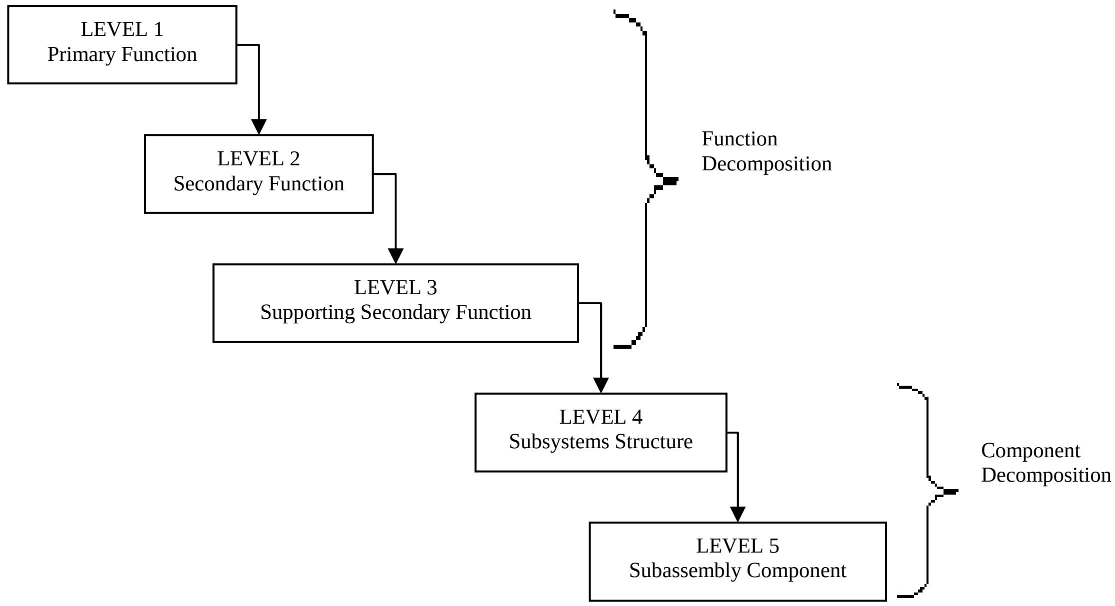

Figure 3. Enhanced functional analysis systems technique system decomposition

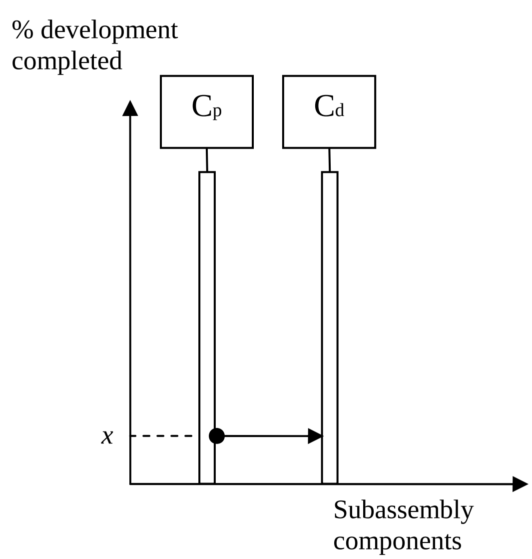

Figure 4. Example of design information flow diagram (DIFD)

Cp	ACT1	ACT2
ACT1	X	Z
ACT2	X	X

% development completed for ACT 1

Figure 5. Example of activity dependency matrix (ADM)

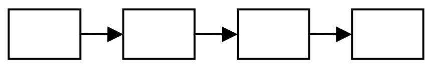

Figure 6. Example of serial network

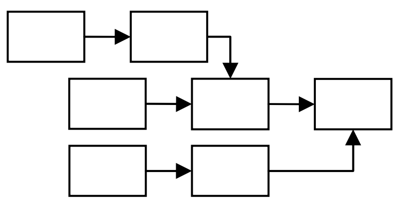

Figure 7. Example of parallel network

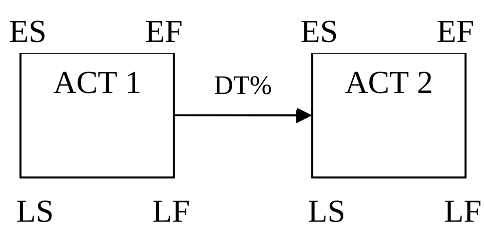

Figure 8. Example of network diagram (ND)

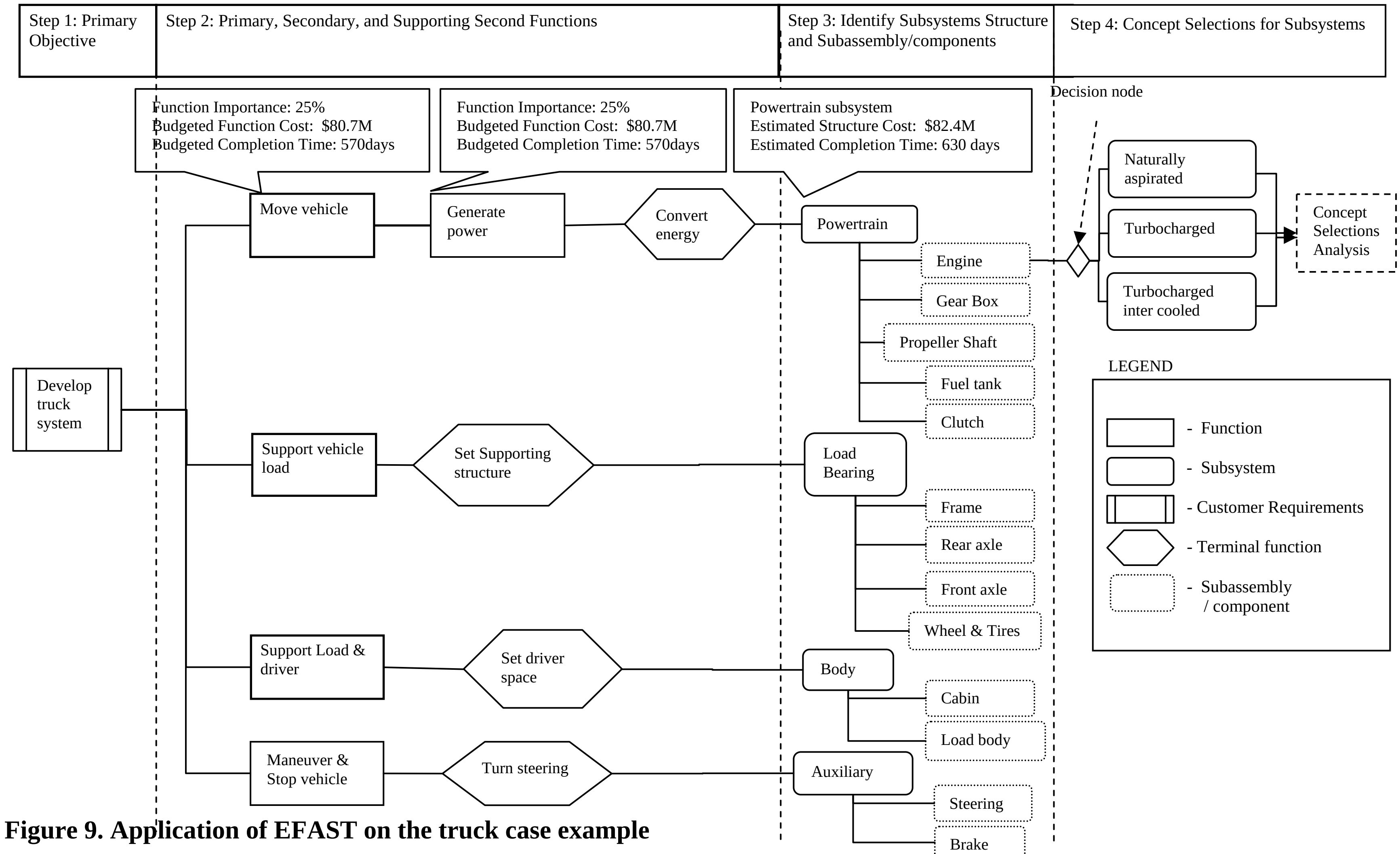

Figure 9. Application of EFAST on the truck case example

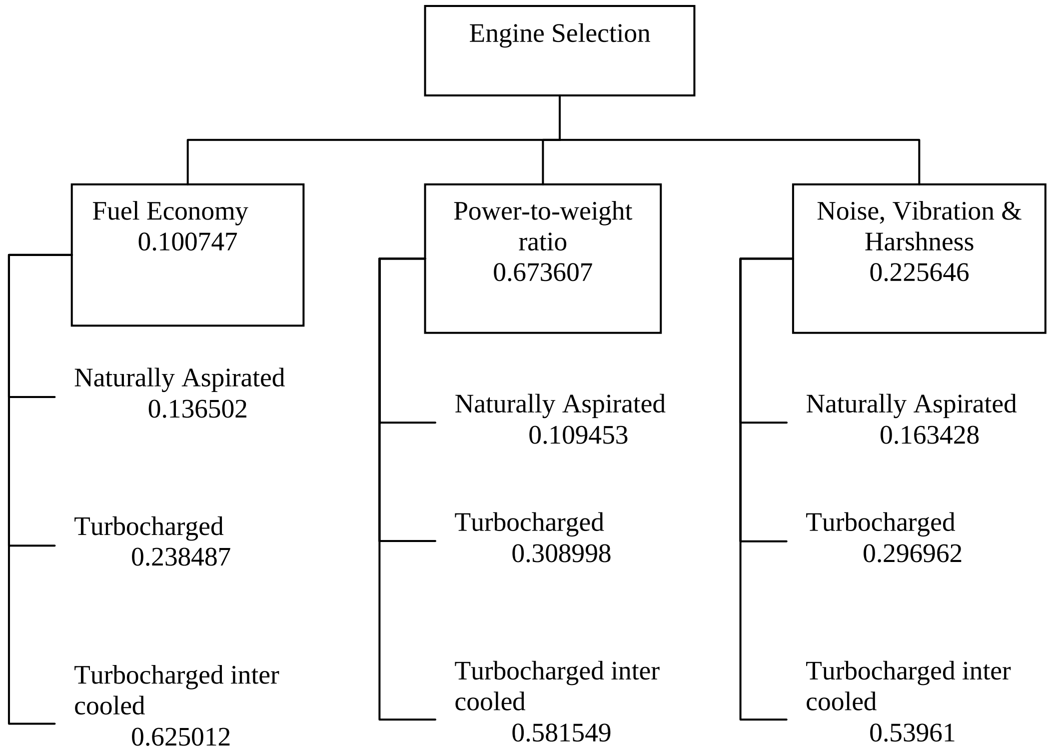

Figure 10. Ranking results from AHP analysis for the truck case example

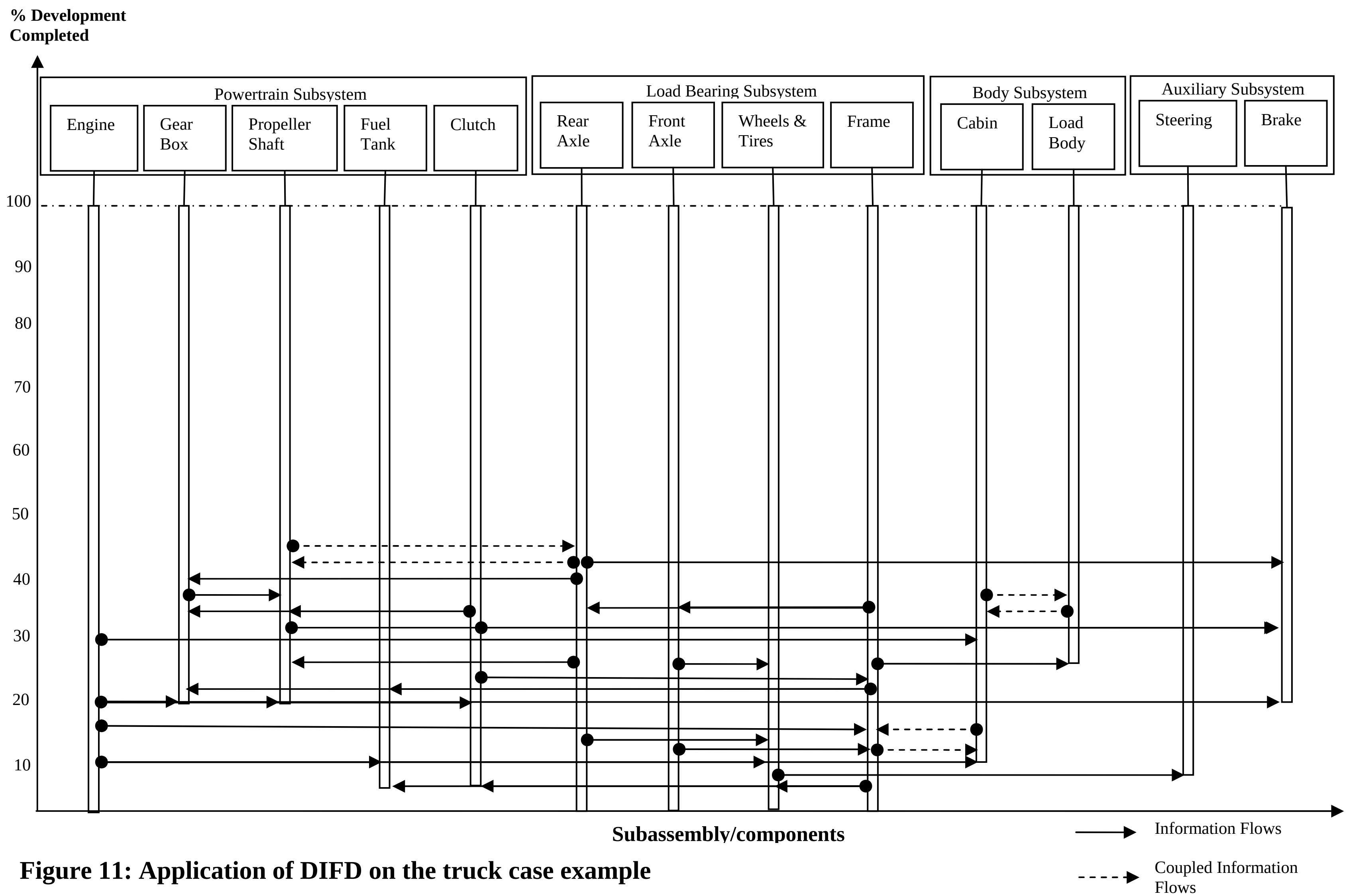

Figure 11: Application of DIFD on the truck case example

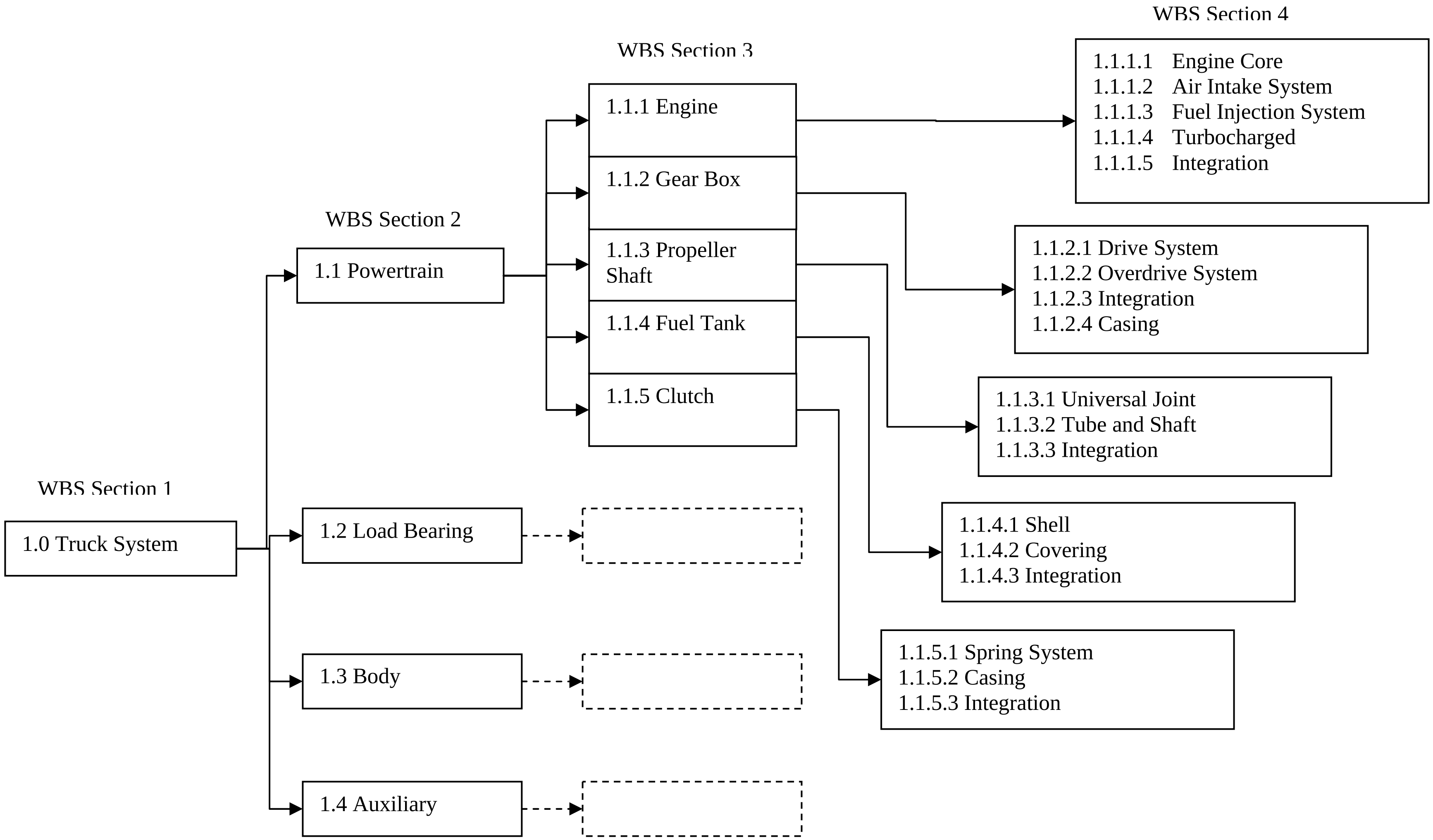

Figure 12. Application of WBS on the truck case example

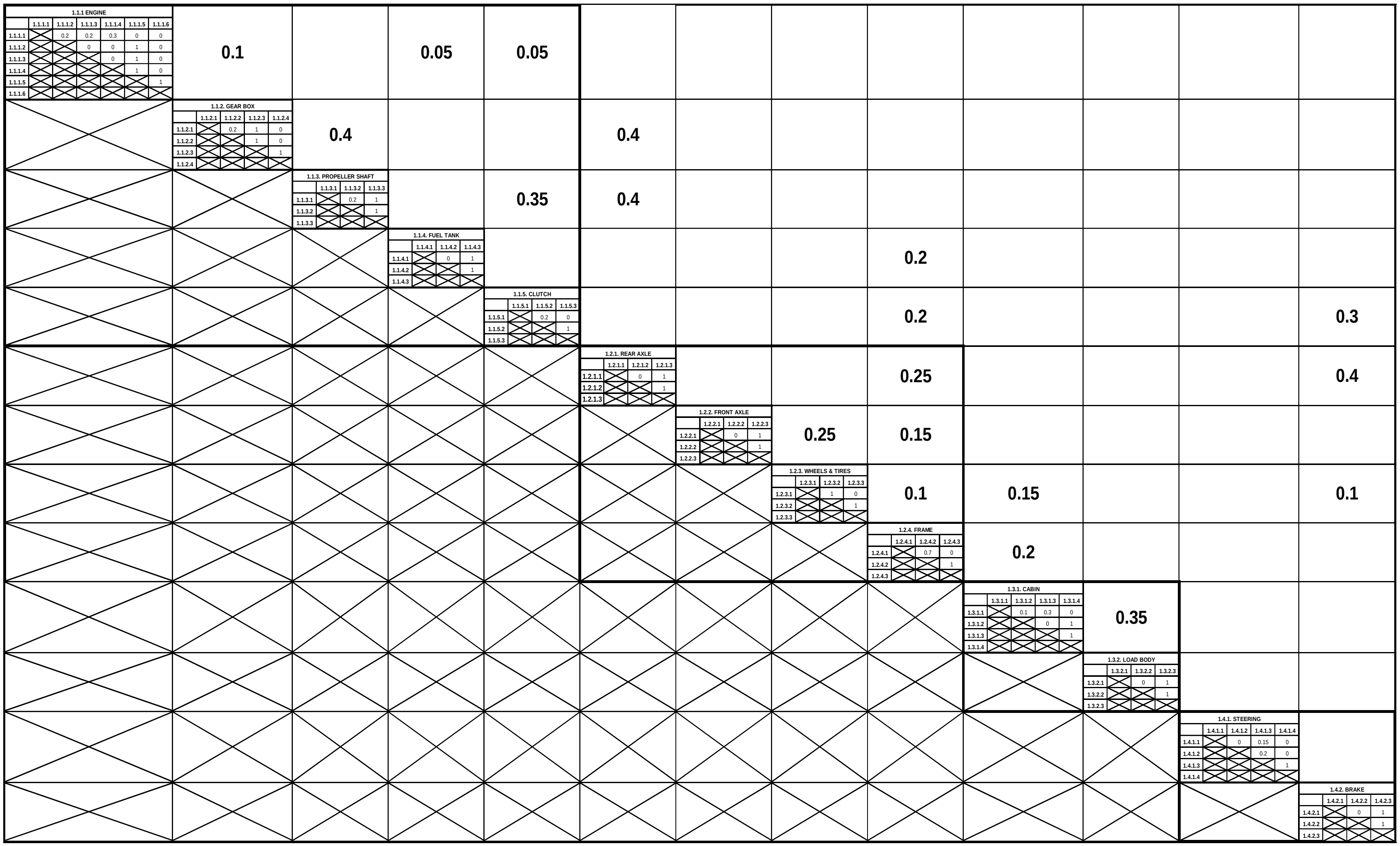

Figure 13. Application of ADM on the truck case example

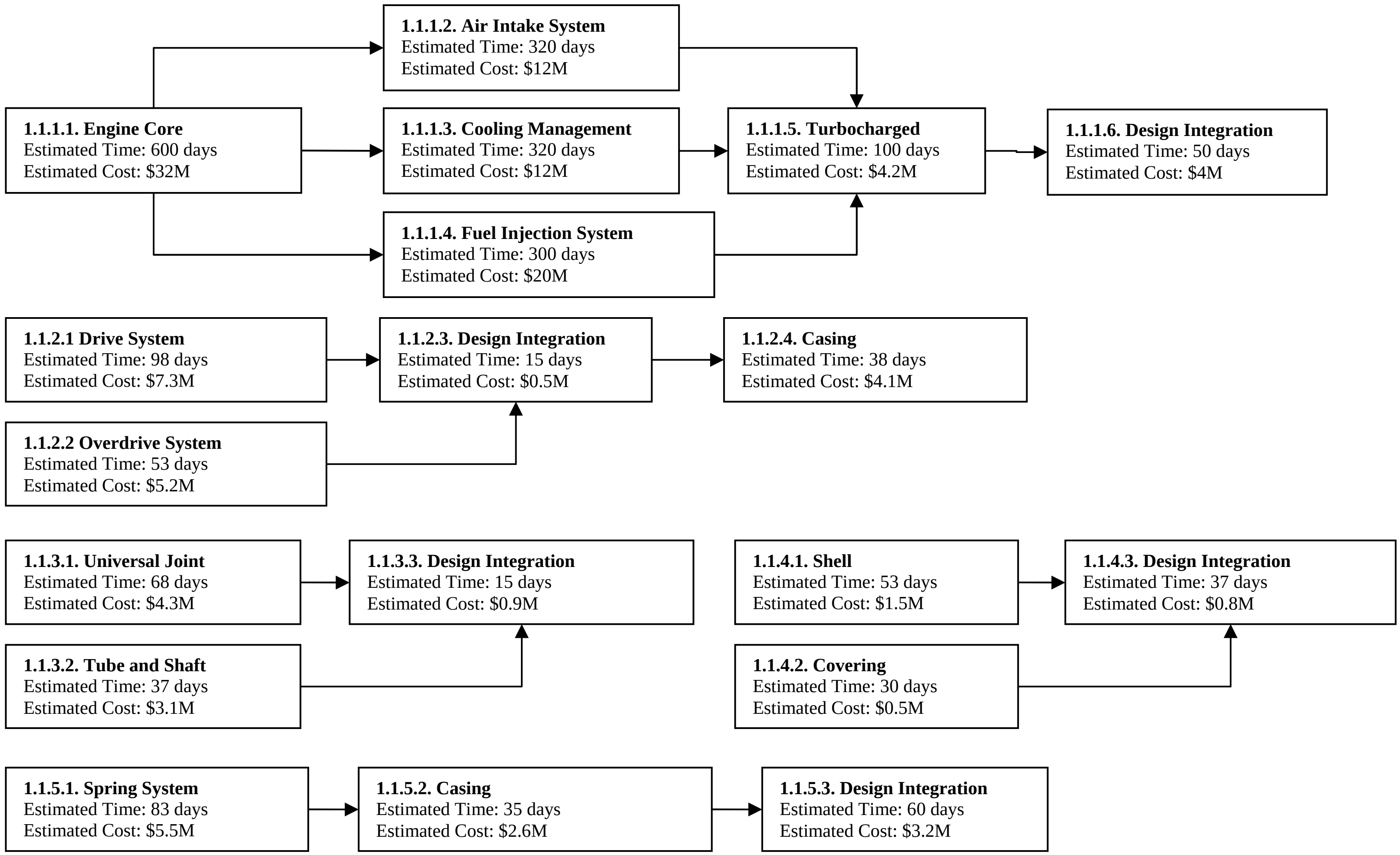

Figure 14. Application of ASD on the truck subsystem case example: Powertrain

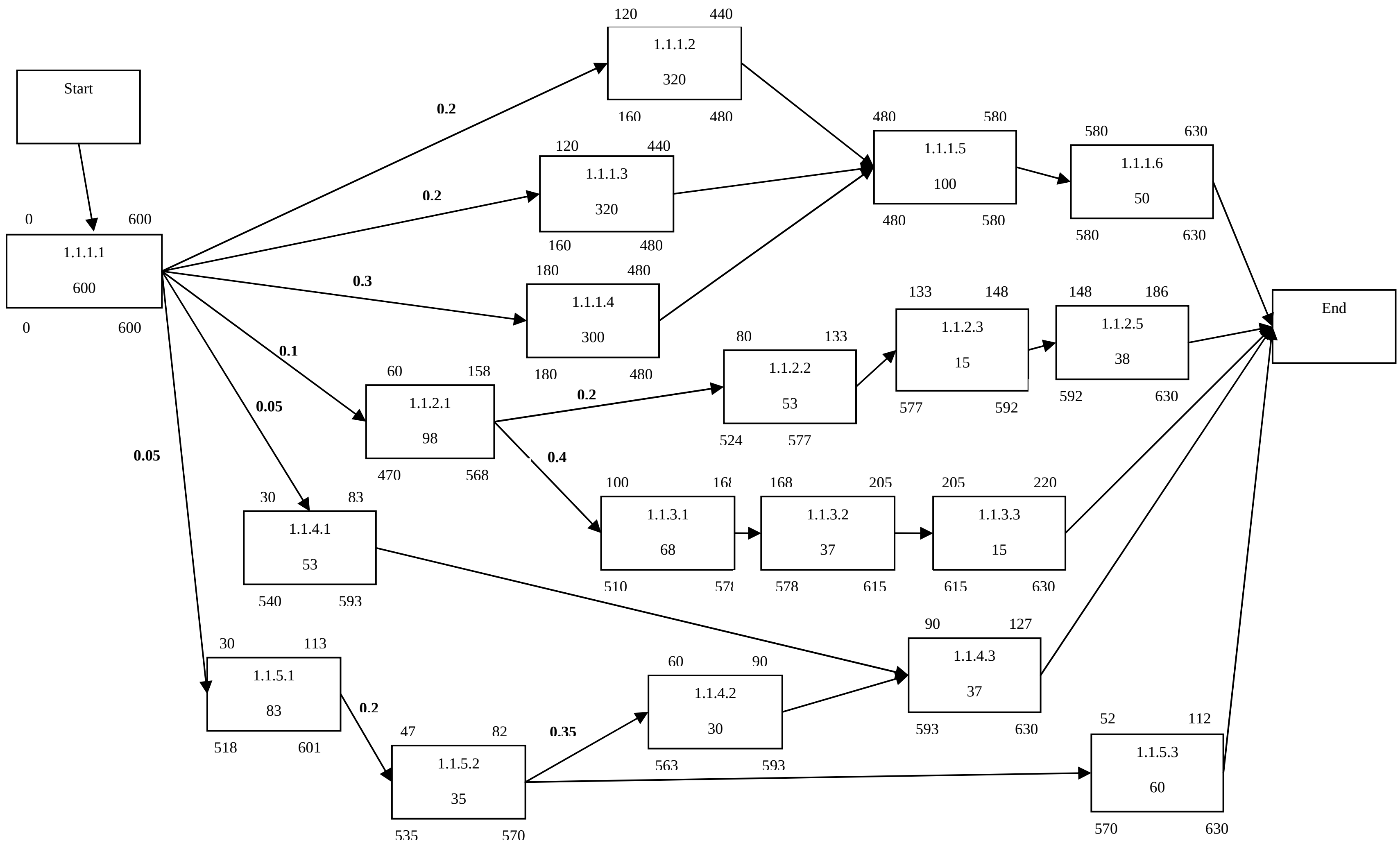

Figure 15. Application of ND on the case example truck subsystem: Powertrain

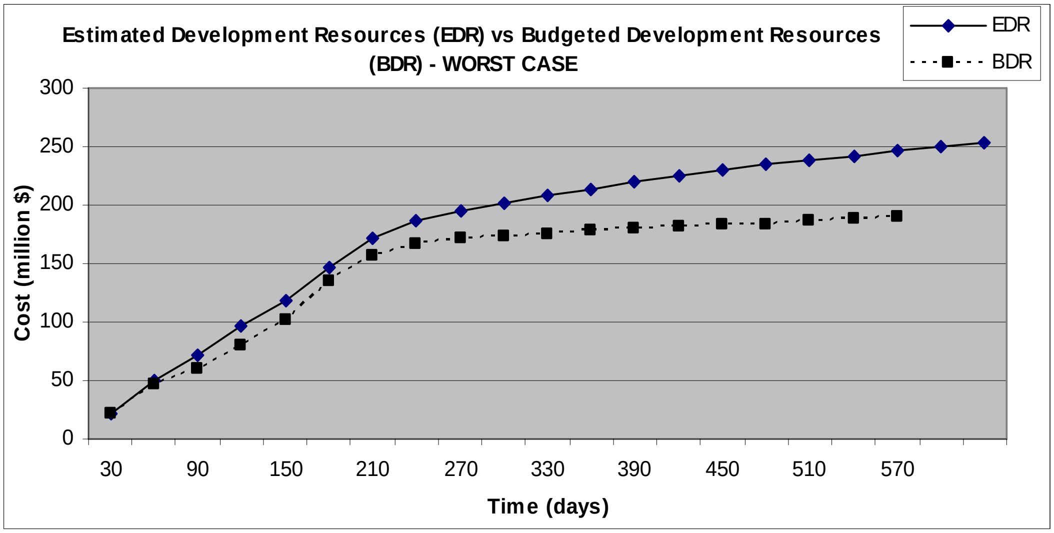

Figure 16. Results for the worst case scenario

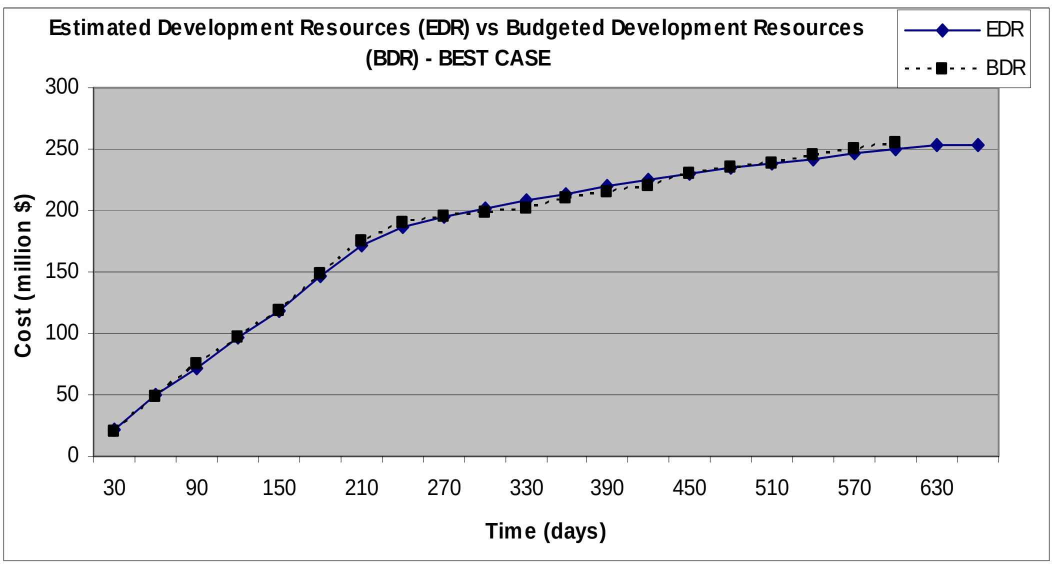

Figure 17. Results for the best case scenario

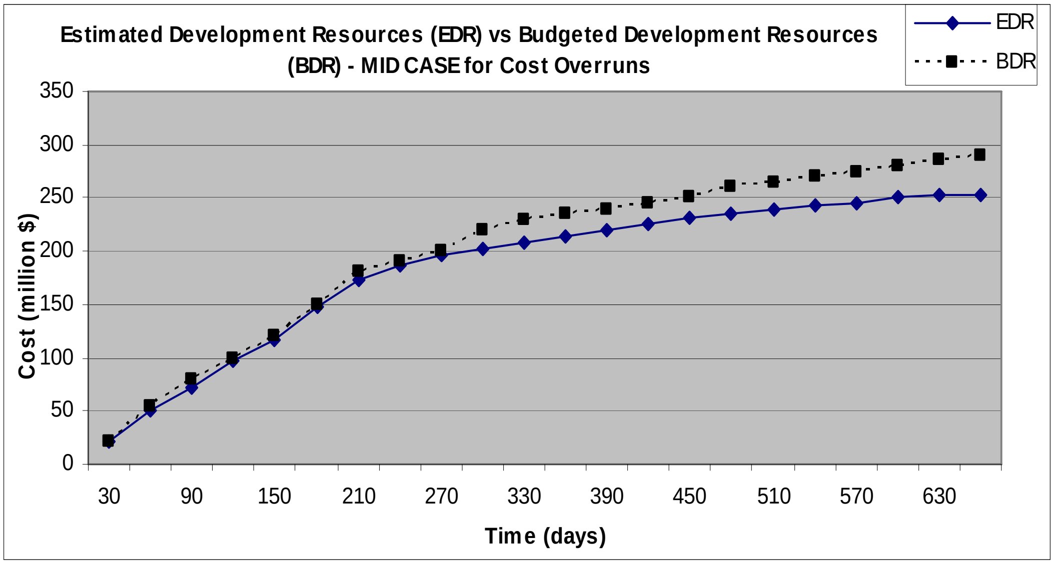

Figure 18. Results for the mid case scenario – cost overrun

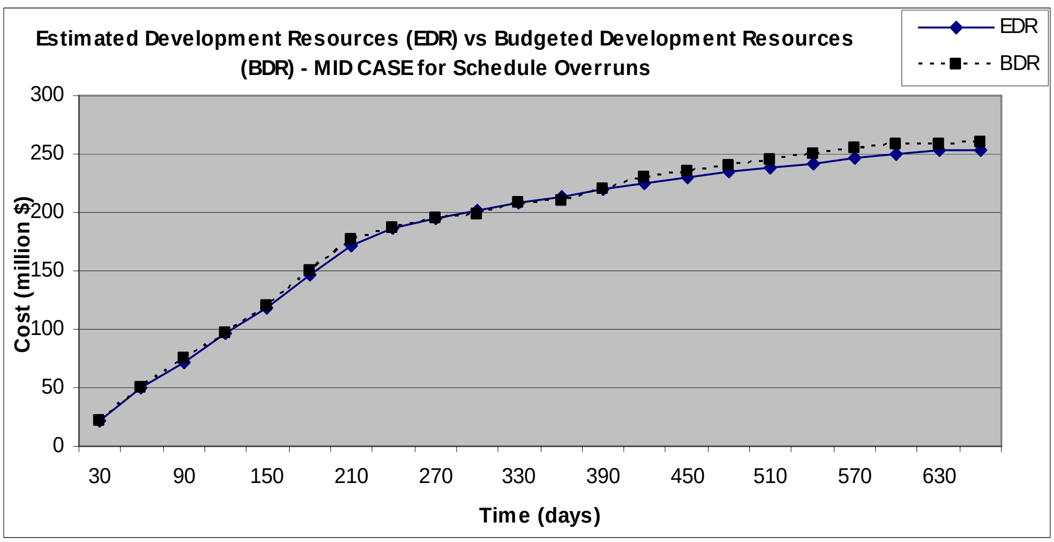

Figure 19. Results for the mid case scenario – schedule overrun

Table I. List of reasons and percentage of contribution to project failures (Standish Group 1995 & 1996 and Scientific American)

Reasons project fails	% contribution to project failure
Incomplete requirements	13.1
Lack of user involvement	12.4
Lack of resources	10.6
Unrealistic expectations	9.9
Lack of executive support	9.3
Changing requirements and specifications	8.7
Lack of planning	8.1
Did not need it any longer	7.5

Table II. List of tools used by systems engineers and project managers (Modified and adapted from INCOSE-TP-2003-016-02, Version 2a).

Systems Engineers Toolkit for Functional Analysis	Project Managers Toolkit
1. Functional Flow Block Diagram 2. N2 charts 3. Timeline Analysis 4. Requirements Allocation 5. Functional Thread Analysis 6. Modeling and Simulation 7. Real-Time Structured Analysis 8. Object-Oriented System Modeling Decision Support 9. Analytic Hierarchy Process 10. Decision Analysis Technique for Risk Management	1. Critical Path Analysis and PERT 2. Gantt Chart 3. Decision Tree 4. Value Analysis 5. Pareto Analysis 6. Cost/Benefit Analysis 7. Stakeholder Analysis

Table III: Engine design parameter ranges for the truck case example

Parameter	Requirement Range
Rotation-per-minute	1400 rpm – 1490 rpm
Torque	892 Nm – 941 Nm
Horsepower of engine (HP)	249 hp – 263 hp

Table IV: Scale for AHP analysis

Scale	Description of the scale
1	Equal
2	Moderate
3	Strong
7	Very Strong
9	Extreme

Table V: Pairwise comparison of design parameter on the truck case example

Design Parameters	Rotation-per-minute	Torque	Horsepower
Rotation-per-minute	1	1/5	1/3
Torque	5	1	4
Horsepower	3	1/4	1

Table VI. The relative ranking of the design parameters on the truck case example

Design Parameters	Computed Eigenvector	Relative ranking
Rotation-per-minute	0.100747	3
Torque	0.673607	1
Horsepower	0.225646	2

Table VII: Pairwise comparison of the alternative concepts engine judged by RPM

Alternative Concept	Naturally Aspirated	Turbocharged	Turbocharged Inter-Cooled
Naturally Aspirated	1	1/5	1/3
Turbocharged	5	1	4
Turbocharged Inter-Cooled	3	1/4	1

Table VIII. The relative ranking of the alternative concepts engine judged by RPM

Alternative Concept	**Computed Eigenvector**	**Relative ranking**
Naturally Aspirated	0.163428	3
Turbocharged	0.296962	2
Turbocharged Inter-Cooled	0.53961	1

Table IX: Pairwise comparison of the alternative concepts engine judged by torque

Alternative Concept	Naturally Aspirated	Turbocharged	Turbocharged Inter-Cooled
Naturally Aspirated	1	1/2	1/3
Turbocharged	2	1	1/2
Turbocharged Inter-Cooled	3	2	1

Table X. The relative ranking of the alternative concepts engine judged by torque

Alternative Concept	**Computed Eigenvector**	**Relative ranking**
Naturally Aspirated	0.163428	3
Turbocharged	0.296962	2
Turbocharged Inter-Cooled	0.53961	1

Table XI: Pairwise comparison of the alternative concepts engine judged by HP

Alternative Concept	Naturally Aspirated	Turbocharged	Turbocharged Inter-Cooled
Naturally Aspirated	1	1/2	1/4
Turbocharged	2	1	1/3
Turbocharged Inter-Cooled	4	3	1

Table XII. The relative ranking of the alternative concepts engine judged by HP

Alternative Concept	**Computed Eigenvector**	**Relative ranking**
Naturally Aspirated	0.136502	3
Turbocharged	0.238487	2
Turbocharged Inter-Cooled	0.625012	1

Table XIII: Estimated cost (EC) and budgeted cost (BC)

COMPONENT	WBS (LEVEL 3)	Estimated Cost (EC), $MIL	Budgeted Cost (BC), $MIL
ENGINE	1.1.1	42.9	40.0
GEAR BOX	1.1.2	17.1	16.1
PROPELLER SHAFT	1.1.3	8.3	7.8
FUEL TANK	1.1.4	2.8	3.8
CLUTCH	1.1.5	11.3	13.0
Total		***82.4***	***80.7***
REAR AXLE	1.2.1	25.2	26.5
FRONT AXLE	1.2.2	24.2	22.1
WHEELS & TIRES	1.2.3	3.1	4.4
FRAME		18.6	17.6
Total		***71.1***	***70.6***
CABIN	1.3.1	8.3	7.5
LOAD BODY	1.3.2	7.4	7.5
Total		***15.7***	***15.0***
STEERING	1.4.1	9.0	10.5
BRAKE	1.4.2	23.4	22.5
Total		***32.4***	***33.0***
OVERALL TOTAL		**201.6**	**199.2**

Table XIV. Estimated time (ET) and budgeted time (BT)

COMPONENT	WBS (LEVEL 3)	Estimated Time (ET), Days	Budgeted Time (BT), Days
ENGINE	1.1.1	600	570
GEAR BOX	1.1.2	193	195
PROPELLER SHAFT	1.1.3	125	130
FUEL TANK	1.1.4	93	95
CLUTCH	1.1.5	106	106
REAR AXLE	1.2.1	225	230
FRONT AXLE	1.2.2	241	240
WHEELS & TIRES	1.2.3	35	35
FRAME		210	205
CABIN	1.3.1	118	120
LOAD BODY	1.3.2	86	85
STEERING	1.4.1	106	105
BRAKE	1.4.2	72	75

VITA

Sofia Tan was born on December 01, 1980 in Alor Setar, Malaysia. Her primary and secondary education was received at St. Nicholas Convent – Alor Setar, Malaysia. In August 1999, she pursued her Bachelors degree in the Electrical Engineering from the University of Missouri – Rolla, Rolla, Missouri, USA. During her undergraduate studies, she was given an opportunity to do her internship with Philips Semiconductor, Zurich, Switzerland. All through the internship, she was involved in several projects related to testing and inspection of electronic architecture. This has inspired her to pursue higher studies in systems architecture. She has been enrolled in the Dept. of Systems Engineering at the University of Missouri – Rolla since January 2006. During this time, she was part of the Sustainable Design Laboratory (SDL) and has handled several projects in the area of Lean Manufacturing. She has attended few conferences in Lean subject and has published a conference paper in the Conference of Systems Engineering Research (CSER). The experiences in her graduate study have given her the confidence and the motivation to continue improving her interpersonal development. She graduated with M.S in Systems Engineering from University of Missouri – Rolla in December 2007.

Sofia enjoys working in a team and solving problems. In her personal life, she loves sport, art, music and plays piano during her leisure time. Apart from that, she be keen on socializing and has been known as a very helpful person to colleagues and her peers.