CRANFIELD UNIVERSITY

NIKOS GIANNOPOULOS

Estimating the Design and Development Cost of Electronic Items

SCHOOL OF APPLIED SCIENCES

PhD THESIS

Cranfield University

School of Applied Sciences

Department of Manufacturing

DOCTOR OF PHILOSOPHY

2006

NIKOS GIANNOPOULOS

Estimating the Design and Development Cost of Electronic Items

Supervisor: Professor Rajkumar Roy

August 2006

THIS THESIS IS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

THIS PAGE IS INTENTIONALLY LEFT BLANK

Abstract

This thesis is concerned with understanding the issues in generating cost estimates at the conceptual design stage for Embedded Systems Design and Development (ES D&D), based on specifications. The research examines if there are any relationships between the system's specifications and the system's cost, and if these relationships can be formalised. The aim is to develop a framework that will structure, formalise and improve the ES D&D Cost Estimating process.

Literature review examines current situation regarding ES D&D Cost Estimating and the information requirements for generating cost estimates. The review identifies that research concentrates on Embedded Systems manufacturing cost estimation, there is a lack of research regarding D&D cost estimation, as well as on the information requirements for generating D&D cost estimates.

By conducting an industrial survey, the author identifies the internal practice on ES D&D Cost Estimating for the automotive and aerospace industries and identifies trends, commonalties and differences within and between them. The survey establishes that in order to improve the ES D&D Cost Estimating process, it is essential to establish a data infrastructure that will avoid issues with shortage of information imposed by suppliers and will link the Embedded System's specifications with the system's actual implementation and expected functionality.

Using a case study approach, the author also establishes that it is essential to analyse the product functionality in such a way that will enable the development of a detailed cost estimating framework at the specification's design stage. The framework is developed in three parts for hardware, software and integration and reuse. The ES hardware design and development effort is predicted using a complexity based cost estimating approach. The research has demonstrated that Use Case Points can be used to predict software development effort for ES software development when the specification is expressed as use cases. In case of statechart based specifications, the development effort is predicted, like in the case of Hardware, using a complexity based cost estimating approach. The study then investigates factors that affect Integration and Reuse effort for ES D&D. The Integration and Reuse effort is predicted using a expert judgement based methodology.

The developed results provide automotive industry with a structured, consistent approach to develop cost estimates for the ES D&D Cost at the specifications design stage. The approach contributes towards improvement of the cost estimating practice within the automotive industry.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Acknowledgements

I would like to take this opportunity to express my thanks to:

My family and my in-laws family members, back in Greece, for their constant support all these years. Especially, a big, loving ''Thank you'' to my wife, Despoina, for her love, support and understanding while I was working on this research. Thank you. I would not have done it without you.

Special thanks to two very special persons and my two best friends: Vaggelis Lavdas and Stathis Kefallonitis. They were my first line of defense and the people that support me throughout all the difficult times. Their help, love and support were invaluable.

To Professor Rajkumar Roy, for his support and guidance throughout my studies at Cranfield University. His support was crucial to my starting, and completing this research.

To the staff of the Department of Enterprise Integration for providing a fantastic and well-facilitated working environment. In particular I would like to thank Mrs. Linda Willsher and Mrs. Tania Pickett, for their always friendly and helpful assistance.

To the Decision Engineering Group members, for offering me the opportunity to join them, for sharing with me their knowledge and opinions. They allowed me to improve throughout all this year my skills and myself.

I would like to acknowledge all the sponsors of the project for their continued support. A special acknowledgement to Terry Griggs and Andy Langridge, for encouragement, support, and genuine interest throughout the research. Without them, this research would never be possible.

Finally, many people have contributed to me getting this far with my PhD. A big thank you goes to everybody else not mentioned above, that somehow supported me in concluding this research. Thank you all so very very much.

THIS PAGE IS INTENTIONALLY LEFT BLANK

List of Publications

Giannopoulos N, Dr. Roy R, and Dr. Taratoukhine V, 'Integration of Enterprise Resource Planning (ERP) and Cost Estimating (CE) systems: The Challenges'. Proceedings of the 18TH National Conference on Manufacturing Research (NCMR), 10th-12th September 2002, Leeds, UK, pp 341-347.

Giannopoulos N, Dr. Roy R, Dr. Taratoukhine V, Sarasua A, and Griggs, T, ‘’Embedded Systems SW Cost Estimating within the Concurrent Engineering Environment’’. Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications (CE2003), Madeira, Portugal, 26-30 July, 2003, pp550-557.

Sarasua A., Roy R., Taratoukhine V. and Giannopoulos N, ‘’Printed Circuit Board Design and Manufacturing Cost Estimation: The Challenges’’. Proceedings of the 1st International Conference on Manufacturing Research (ICMR), Glasgow, Scotland, 9-11 September, 2003, pp 258-265.

Giannopoulos N, Roy R., Taratoukhine V. and Sarasua A., ‘’Embedded Software Cost Estimating in the Automotive Industry: Cranfield’s Approach’’. Proceedings of the 11th ISPE International Conference on Concurrent Engineering: Research and Applications (CE2004), Beijing, China, 26-30 July, 2004, pp 452-459.

Giannopoulos N, Roy R., Panetto H, Nunez M. J., Roca de Togores A, ‘’Web Services : the interoperability solution in Extended/Virtual enterprises’’. Proceedings of the 11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), Salvador, Bahia, Brazil, 5-7 of April, 2004, pp 235-242

Sarasua A., Roy R., Taratoukhine V. and Giannopoulos N, "Design and development cost for printed circuit boards", Proceedings of the 2nd International Conference on Manufacturing Research (ICMR), Sheffield Hallam University, 7-9 September, 2004, pp 387-394

Giannopoulos, N., and Roy, R., ‘’ Embedded Software Design & Development Effort Estimation in the Automotive Industry Based on Statecharts Specifications’’, submitted to the IEE Transactions on Software, 10 of August, 2006

THIS PAGE IS INTENTIONALLY LEFT BLANK

Table of contents

ABSTRACT 4
ACKNOWLEDGEMENTS 6
LIST OF PUBLICATIONS 8
LIST OF FIGURES 14
1. INTRODUCTION 19
1.1. RESEARCH BACKGROUND 19
1.2. ELECTRONICS COST ESTIMATING WITHIN AUTOMOTIVE INDUSTRY 26
1.3. RESEARCH CONTEXT 27
1.4. INDUSTRIAL COLLABORATOR: FORD MOTOR COMPANY 28
1.5. COST ESTIMATING AT FORD MOTOR COMPANY 29
1.6. RESEARCH AIM 30
1.7. THESIS STRUCTURE 30
2. LITERATURE REVIEW 33
2.1. EMBEDDED SYSTEMS DESIGN AND DEVELOPMENT 33
2.1.1: Traditional Design 34
2.1.2: Co - Design 36
2.1.3: System-level design and model-based design 38
2.1.4: General observations on model based design 44
2.1.5: Co – Design with IPs 45
2.1.6: Co – Design with Reusable components 48
2.1.7: Latest developments in Embedded Systems D&D 50
2.1.8: Complexity 54
2.2. EMBEDDED SYSTEMS IN THE AUTOMOTIVE INDUSTRY 55
2.2.1. SW to SW 58
2.2.2. SW to HW 61
2.2.3. Embedded system in the car's electronic architecture 63
2.3. EMBEDDED SYSTEMS D&D IN OTHER INDUSTRIES 65
2.4. EMBEDDED SYSTEMS SPECIFICATION APPROACHES 68
2.4.1. Use case based specifications 68
2.4.2. Statecharts based specifications 70
2.5. EMBEDDED SYSTEMS COST ESTIMATION 72
2.5.1. HW 74
2.5.2. SW 79
2.5.3. Reuse 90
2.5.4. Intres 90
2.6. ESTIMATING D&D COST 91
2.7. GAP ANALYSIS 97
2.8. SUMMARY AND KEY OBSERVATIONS 97
3. RESEARCH AIM, OBJECTIVES AND METHODOLOGY 101
3.1. RESEARCH AIM AND OBJECTIVES 101
3.2. OVERVIEW OF RESEARCH METHODOLOGY 103
3.2.1. Research Purpose 103
3.2.2. Selecting a Quantitative or Qualitative Research Approach 104
3.2.3. Research Strategy 105
3.3. CASE-STUDY STRATEGY 106
3.3.1. Case Studies and Data Collection 106

3.3.2. Case Study Issues ... *107*
3.4. RESEARCH METHODOLOGY ... 109
3.5. SUMMARY ... 112

4. AS-IS MODEL CAPTURE ... 113

4.1. DESIGN OF AS-IS STUDY ... 113
4.1.1. Target Audience ... *113*
4.1.2. Questionnaire Development ... *114*
4.1.3. Conducting the Interviews ... *115*
4.2. ANALYSIS OF AS-IS STUDY ... 117
4.2.1. Performing the analysis of results ... *117*
4.2.2. Embedded Systems CE in the first automotive Organisation ... *120*
4.2.3. Presentation of results of the automotive companies ... *128*
4.2.4. Presentation of results of aerospace companies ... *129*
4.2.5. Comparison of results of automotive and aerospace companies ... *132*
4.2.6. Results validation ... *133*
4.2.7. General observations ... *133*
4.3. KEY OBSERVATIONS FROM THE RESULTS. ... 135
4.4. SUMMARY ... 136

5. TO – BE PROCESS DEVELOPMENT ... 137

5.1. DESIGN OF THE TO-BE STUDY: CAPTURING D&D EFFORT ... 137
5.1.1. Target Audience ... *137*
5.2. RESULTS ... 143
5.2.1. Performing the analysis of results ... *143*
5.2.2. Results verification ... *143*
5.2.3. Presentation of results ... *143*
5.3. DEVELOPMENT OF THE TO-BE MODEL ... 145
5.3.1. TO – BE model validation ... *147*
5.4. AS – IS VS TO - BE ... 149
5.5. SUMMARY ... 153

6. DEVELOPING A COST ESTIMATING FRAMEWORK FOR EMBEDDED HARDWARE DESIGN AND DEVELOPMENT ... 154

6.1. DEVELOPING THE COMPLEXITY METRIC ... 155
6.2. METRIC VALIDATION AND WEIGHT ALLOCATION ... 158
6.3. CASE STUDIES ... 165
6.3.1. Presentation of the results ... *169*
6.4. FRAMEWORK VALIDATION ... 169
6.4.1. Presentation of the results ... *171*
6.5. SUMMARY ... 172

7. DEVELOPING OF A COST ESTIMATING FRAMEWORK FOR EMBEDDED SW DESIGN AND DEVELOPMENT ... 173

7.1. USE CASE BASED EFFORT ESTIMATION: CASE STUDIES ... 174
7.2. STATECHARTS BASED METRICS AND EFFORT ESTIMATION ... 179
7.2.1. Case Studies ... *185*
7.3. METRICS AND RESULTS VALIDATION ... 189
7.3.1. Workshop design ... *189*
7.3.1.1. Target Audience ... 189
7.3.1.2. Questionnaire Development ... 190
7.3.1.3. Conducting the Workshop ... 190
7.4. SUMMARY ... 192

8. UNDERSTANDING REUSE AND INTEGRATION 193
8.1. DEVELOPING A REUSE ESTIMATION FRAMEWORK 193
8.1.1. Conducting the interviews 196
8.1.2. Results Presentation 201
8.1.3. Results Validation 202
8.2. INTEGRATION 204
8.2.1. Introduction 204
8.2.2. Developing an approach for improving Itegration's Visibility. 204
8.2.2.1. Presentation of the results 209
8.2.3. Results validation 209
8.2.3.1. Presentation of the results 211
8.3. SUMMARY 211
9. FRAMEWORK IMPLEMENTATION AND VALIDATION 212
9.1. FRAMEWORK PRESENTATION 212
9.2. FRAMEWORK VALIDATION 222
9.2.1. Workshop Design 222
9.2.1.1. Target Audience 222
9.2.1.2. Conducting the Interviews 223
9.2.2. Results Presentation 230
9.3. SUMMARY 231
CHAPTER 10: DISCUSSION AND CONCLUSIONS 232
10.1. INTRODUCTION 232
10.2. KEY RESEARCH OBSERVATIONS 232
10.2.1. Literature Review 232
10.2.2. AS-IS 233
10.2.3. AS – IS versus TO – BE 235
10.2.4. Framework Development 236
10.2.4.1. SW Module 236
10.2.4.2. HW Module 237
10.2.4.3. Reuse and Integration Module 238
10.2.4.4. General Comments 238
10.2.5. Framework Validation 239
10.3. RESEARCH CONTRIBUTIONS 240
10.4. IMPLEMENTATION ISSUES 241
10.4.1. Technical Issues 241
10.4.2. Maintenance Issues 242
10.4.3. Financial Issues 242
10.4.4. Cultural Issues 243
10.5. LIMITATIONS 243
10.6. FUTURE RESEARCH 245
10.7. CONCLUSIONS 246
REFERENCES 248
APPENDICES 268
APPENDIX A : AS - IS QUESTIONNAIRE 269
APPENDIX B : HEAD OF ELECTRONICS DESIGN 274
APPENDIX C : D&D ELICITATION QUESTIONNAIRE 278
APPENDIX D : STATECHARTS COMPLEXITY METRIC 283
APPENDIX E : SW METRICS AND RESULTS 288
APPENDIX F : COMPLEXITY QUESTIONNAIRE 293
APPENDIX G : HW D&D COST ESTIMATION 299
APPENDIX H : REUSE ESTIMATION FRAMEWORK 303

APPENDIX I : INTEGRATION BRAKE – DOWN 308
APPENDIX J : REUSE ESTIMATION VALIDATION 312
APPENDIX K : INTEGRATION VALIDATION QUESTIONNAIRE 316
APPENDIX L : TRANSCRIPT 321
APPENDIX M : IDEF 335

List of Figures

1.1: An embedded system and the environment	**20**
1.2: Resources encompassed in an Embedded System	**22**
1.3: Cost Estimating within Ford	**29**
1.4: Thesis structure	**32**
2.1: Traditional Design Flow	**35**
2.2: Shorter time to market using co-design	**36**
2.3: Commonalties in SW and HW development teams	**37**
2.4: Model-based co-design flow	**40**
2.5: Design flow using IPs	**47**
2.6: A typical car ECU	**53**
2.7: A typical ECU	**53**
2.8: The automotive electric system	**56**
2.9: ‘’V’’ design flow	**57**
2.10: A typical car ECU	**58**
2.11: Strategies for either components integration	**60**
2.12: HW/SW integration	**63**
2.13: A Use Case diagram example	**68**
2.14: Example of use case documentation	**70**
2.15: A statechart diagram	**71**
2.16: Anatomy of detailed estimate	**75**
2.17: Example of a parametric equation	**76**
2.18: Increasing car electronics complexity	**79**
2.19: Cost commitment over the product lifecycle	**91**
3.1: Research Methodology Map	**111**
4.1: IDEF$_0$ Syntax Representation	**121**
4.2: Function Decomposition Using IDEF$_0$ Notation	**122**
4.3: 1st automotive company ES cost estimating process	**123**
4.4: Relationships between individual IDEF0 Diagrams	**124**
4.5: 2nd automotive company ES cost estimating process	**129**
4.6: 1st Aerospace company (company 3) Electronic Items Cost	**131**
4.7: 2nd Aerospace company (company 4) Electronic Items Cost	**131**
5.1: Distribution of electronic items D&D effort	**144**
5.2: TO – BE model	**146**
5.3: IDEF$_0$ representation for the ES D&D TO – BE model	**148**

5.4: A step by step approach to ES Cost Estimation	**149**
5.5a: TO-BE model ES Cost Estimation – Manufacturing Cost	**150**
5.5b: TO-BE model ES Cost Estimation – Mark Up Costs	**151**
5.5c: TO-BE model ES Cost Estimation – D&D Cost Estimation	**152**
5.5d: TO-BE model ES Cost Estimation – Piece Price Estimation	**153**
6.1: Linking Complexity to HW D&D Cost	**154**
6.2: IP Protection Classes	**162**
6.3: Complexity vs HW D&D cost	**168**
7.1: A Use Case diagram	**174**
7.2: A Use Case documentation	**175**
7.3: A statechart diagram	**181**
7.4: Parallel machines	**182**
7.5: Statecharts specification structure	**186**
7.6: Overall Statechart diagram for Case Study 1	**186**
7.7: Statechart diagram for activity 1.1.1 for Case Study 1	**187**
9.1: ES D&D Effort Estimation Framework Initial Screen	**213**
9.2: Initial Screen of the HW D&D Effort Estimation Module: An Example	**214**
9.3: HW D&D Effort Estimation Module After Introducing Values: An Example	**215**
9.4: SW D&D Effort Estimation Module (Use Case Based)	**217**
9.5: SW D&D Effort Estimation Module (Statecharts Based)	**217**
9.6: SW D&D Effort Estimation (Use Case Based): An Example	**218**
9.7: SW D&D Effort Estimation (Statecharts Based): An Example	**218**
9.8: Reuse Effort Estimation Module	**219**
9.9: Reuse Effort Estimation	**220**
9.10: Integration Effort Estimation Module	**221**

List of Tables

1.1: Embedded Systems definition	21
1.2: Hard real-time versus Soft real-time	24
2.1: Reuse Metrics	49
2.2: Characteristics of the strategies for either HW or SW components integration	60
2.3: Characteristics comparison for SW functionality	86
2.4: Actor's classification and weighting	87
2.5: Use Cases' classification and weighting	87
2.6: Technical Factors	88
2.7: Environmental Factors	89
2.8: Design and Development effort estimation models	96
3.1: The purposes of research	103
3.2: Characteristics of research strategies	105
3.3: Strategies for Dealing with Risks to Research Validity	108
4.1: Workshops and Interviews participants	114
4.2: Summary of experts' answers to key questions	118
5.1: Workshop participants	138
5.2: A summary of the participants' observations	141
5.3: D&D effort distribution according to engineers	143
5.4: Table 5.3: TO – BE model's ICOMs	147
6.1: Initial Complexity Driver	157
6.2: Survey participants	158
6.3: Weighting allocation and validation	160
6.4: Amended weighted HW D&D Complexity coefficients	164
6.5: An example of a BOM (Case Study 1)	165
6.6: Complexity factor values for case study 1	166
6.7: Complexity factor values for the 6 case studies	167
6.8: Classification of case studies in descending order according to their complexity value	168
6.9: Interview participants' details	169
6.10: A summary of the experts' answers	171
7.1: Use Case classification for case study 4	176
7.2: Environmental factors calculation for Case Study 4	177
7.3: UCP count for the 7 case studies	178
7.4: Estimation of D&D effort for UML Use Cases	179

Specifications	
7.5: Statechart metrics development experts	180
7.6: Survey participants	183
7.7: A summary of the participants' answers	184
7.8: Design and development effort estimation for Case Study 1	188
7.9: Design and development effort estimation for all Case Studies	188
7.10: Design and development effort estimation	189
7.11: Survey participants	189
7.12: A summary of the workshop participants' answers	191
7.13: Amended Design and development effort estimation	192
8.1: Reuse Framework Matrix	195
8.2: Experts details	196
8.3: Summary of the experts' answers	199
8.4: reuse percentages provided by the experts	200
8.5: Reuse matrix for the 3 case studies	201
8.6: Survey participants' details	202
8.7: Summary of the experts' answers	203
8.8: Details of experts	205
8.9: Integration brake-down matrix	207
8.10: A summary of the experts' answers	208
8.11: Survey participants' percentages	209
8.12: Survey participants' details	209
8.13: A summary of the expert's answers	210
9.1: First OEM Workshop Participants	222
9.2: Second OEM Workshop Participants	222
9.3: OEM 1 Experts Opinions' Summary for HW Validation	224
9.4: OEM 1 Experts Opinions' Summary for Reuse Validation	225
9.5: OEM 1 Experts Opinions' Summary for Integration Validation	226
9.6: OEM 2 Experts Opinions' Summary for HW Validation	227
9.7: OEM 2 Experts Opinions' Summary for Reuse Validation	228
9.8: OEM 2 Experts Opinions' Summary for Integration Validation	229

List of Abbreviations

ASCM - Adjusted Statecharts Complexity Metric

ASW – Application Software

BOM - Bill of Material

CERs - Cost Estimation Relationships

COSMIC FFP - COSMIC Full Function Points

COCOMO – Constructive Cost Model

CPU - Central Processing Unit

D&D – Design and Development

ECU - Electronic Control Unit

ES – Embedded System

FP - Function Points

HDL - High Description Language

HW – Hardware

IC - Integrated Circuit

ICOMs – Inputs, Controls, Outputs, Mechanisms

IDEF - Integrated Definition Function Modelling

IP – Intellectual Property

LCC - Life Cycle cost

LOC - Lines of Code

MarkII - Mark II Function Points

MoC - Models of Computation

MOE - Modular Optimisation Environment

OEM - Original Equipment Manufacturing

SW – Software

SLL - System Level Language

USCM - Unadjusted Statecharts Complexity Metric

UML - Unified Modelling Language

UCP - Use Case Points

1. Introduction

The purpose of this chapter is to provide a foreword to the thesis by introducing the research topic. The context of the research will be described and the problem statement will be presented. The importance of providing a way for linking Embedded System (ES) specifications with its Design and Development (D&D) cost will be established.

Following the description of the problem under investigation, the specific aim of the research will be identified. Consideration will then be given to the scope of the research. Finally the structure of the thesis will be described in order to provide the reader with a general overview.

1.1. Research Background

Cost estimating is not confined to any particular industry or group of industries. Enterprises perform cost estimating throughout the life cycle of a product since any "go'' or ''no-go" decision is based on the accuracy of a background cost estimate.

AACE defines *Cost Estimation* as: "The determination of quantity and the predicting or forecasting, within a defined scope, of the cost required to construct and equip a facility, to manufacture goods, or to furnish a service. Costs are determined utilising experience and calculating and forecasting the future cost of resources, methods and management within a scheduled time frame. Included in these costs are assessment and an evaluation of risks and opportunities. Cost estimation provides a basis for feasibility studies, business planning, budget preparation, and cost and scheduling control" (AACE, 2000). This thesis is concerned with ES D&D cost estimating on the early stage of the product specification within the automotive industry, and its focus is on producing and delivering a model to estimate the cost of ES D&D in that stage.

During recent years, cars undergo a significant change from being primarily mechanical devices into electronic wonders (Cronenweth, 2004). Until recently, the electric system of the car was concerned with lights, starter motors, windshield wipers, etc. The situation started to change when car manufacturers started introducing embedded systems to address requirements like safety (ie ABS, Airbag), information to the driver (ie diagnostics), comfort (ie climate control) etc. (Axelsson, 2002) and today electronics are essential to control the car movements, its emissions, to entertain the passengers, etc (Sangiovanni-Vincentelli, 2000). According to Daimler-Chrysler, ''electronics will be the source for more than 80% of the innovations in the automotive industry'' (Sangiovanni-Vincentelli, 2000).

The growing needs of clients and society for functionality, dependability and environmental issues (driving comfort, fuel efficiency, reduced emissions, etc) together with the decreasing cost of microelectronics led the automobile manufacturers to implement more and more functions of a car using electronic devices, which has opened the important new market of automotive electronics (Kopetz, 1995; Sangiovanni-Vincentelli, 2000). Some cars today contain more than 100 microcontrollers, with many of them working interdependently (Cronenweth, 2004); for example, VOLVO S80 includes ''18 ECUs (Electronic Control Units) connected via six networks: a low speed body electronics CAN bus (125kbit/s), a high speed powertrain CAN bus (250kbit/s and four other networks'' (Cornu and Kung, 2002).

These electronic devices are embedded in a greater architecture (ie the ABS device is embedded on the car) (Sciuto, 2000) and their mission is to monitor and control the operation of the device they are built in (for example, ABS monitors and controls the car's braking process) (Axelsson, 1997a; Lee, 2000; Heath, 2003). For that reason, these electronic devices are also called embedded systems (ES) and this greater architecture consists their environment (Axelsson, 1997a). ES monitor their environment using their sensors, and if the need occurs, they control it using their actuators (Wirth, 2001; Lavagno et al, 1998; Axelsson, 1997a). Under this view, for the scope of this research, the terms electronic item and embedded system (ES) are going to be used interchangeably.

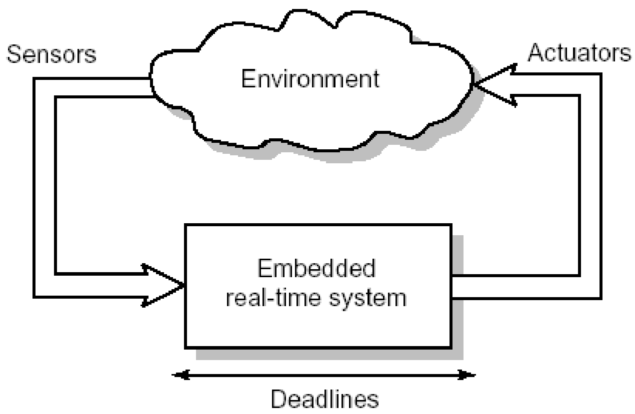


Figure 1.1: An embedded system and the environment (Axelsson, 1997a)

There have been various definitions regarding Embedded Systems:

Table 1.1: Embedded Systems definitions

''An embedded system is a microprocessor based system that is built to control a function or range of functions and it is not designed to be programmed by the end user in the same way a PC is; it can not run excel now and word latter. An embedded system is designed to perform one particular task albeit with choices and different options''	(Heath, 2003)
''Such systems, which use a computer to perform a specific function but are neither used or perceived as a computer, are generally known as embedded systems''	(Edwards et al, 1997)
''An embedded system is a complex object containing a significant percentage of electronic devices (generally including at least one computer) that interacts with the real world (physical environment, human users, etc.) through sensing and actuating devices''	(Lavagno et al, 1998)
''An embedded system is a system whose principal function is not computational, but which is controlled by a computer embedded within it. The word embedded implies that it lies inside the overall system, hidden from view, forming an integral part of a greater whole''	(Wilmshurst, 2001)
''An ES is a microcontroller based, software driven, real time control system, autonomous or human or network interactive, operating on diverse physical variables and in diverse environments, and sold into a competitive and cost conscious market''	(Wilmshurst, 2001)
''An embedded system employs a combination of hardware and software ('a computational engine'') to perform a specific function; it is part of a larger system that may not be a 'computer'; works in a reactive and time constraint environment. Software is used for providing features and flexibility, whereas hardware (processors, ASICs, etc.) is used for performance and sometimes security''	(Gupta, 2003)
''An embedded system is a computer, which is a subsystem of a greater technical system, and is in charge of controlling and operating the entire device. The surrounding world is called 'the environment'. The embedded system typically collects information from the environment, using sensors, and affects it using actuators'	(Axelsson, 1997a)
''An ES is part of a larger whole that typically consists of many components, not just computer modules, but also sensors and actuators. This implies that many activities occur concurrently, and are to be controlled by the computer system part''	(Wirth, 2001)

Observing the definitions presented above and based on their commonalties and differences, it can be concluded that *an embedded system is a complex system consisting of both software and hardware, which is part of a greater architecture* *and its role is to monitor using its sensors the environment within it operates and to control it, using its actuators.*

In this process of observing and controlling their environment, the coordinator is the software (Sastry et al, 2003), which provides the functionality and the flexibility required (Ziebart, 1991; Gupta, 2003), whereas hardware (processors, ASICs, etc.) is used as the underlying computational platform (Gupta, 2003).

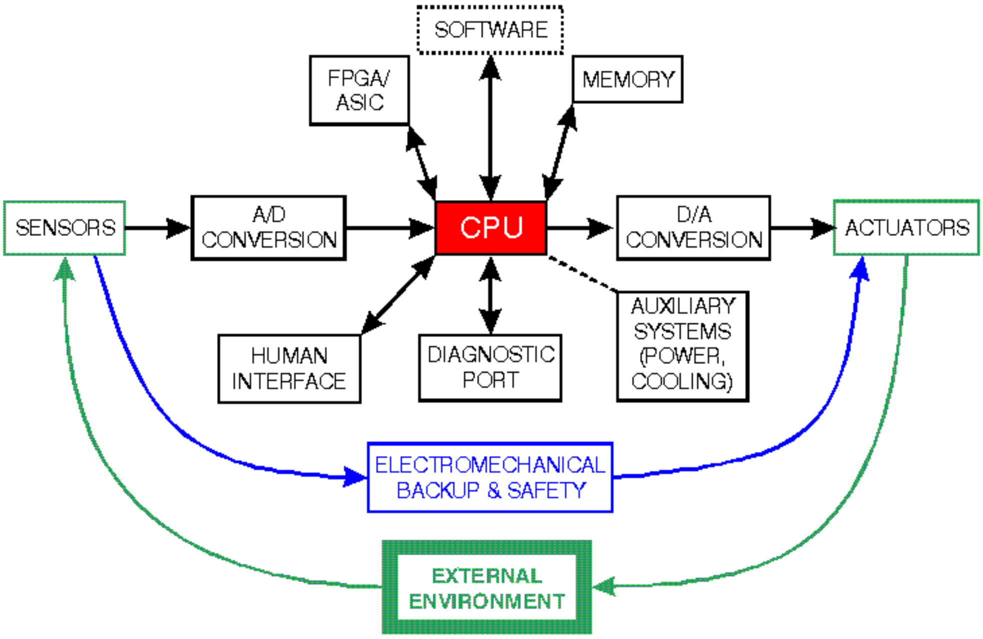


Figure 1.2: Resources encompassed in an Embedded System (Koopman, 1996)

ES have been created to address a specific task and for that reason they are customised in performing it (Koopman, 1996; Axelsson, 1999; Berger, 2002; Gupta 2002; Vahid and Givargis, 2002; Axelsson, 1997a). That means that each ES is a unique case (Koopman, 1999), however the following general characteristics could be identified (Wilmshurst, 2001; Wirth, 2001; Gupta, 2003; Kopetz, 1997; Lee, 2000; Berger, 2002; Koopman, 1999):

- It is highly unlikely for an ES to respond to only one single event; they receive inputs from both the network and the environment through their sensors and they have to concurrently control both their network and their environment through their actuators.

- ES should be up and running all the time; so, they must contain no bugs and it should not, under no circumstances, halt or terminate their operations.
- ES are produced in mass quantities and for that reason they are cost sensitive.
- ES are embedded in a greater architecture and access to them is most of the times difficult.
- They are of high quality standards, since after they are manufactured it is very difficult to be altered.
- They are difficult to maintain, since the access to them after they are embedded to the greater architecture is in most of the times difficult.
- When the software that is contained in the ES (embedded software) fails, this could lead to big catastrophes, unlikely of the desktop software
- ES have constraints in their power consumption
- ES operate under any circumstances and sometimes in environments that are unapproachable by humans

As with the definitions, many categorisations regarding ES can be found in literature:

(i) Soft real-time vs Hard real-time:

- **Real time:** the ES has to produce its response within a specific and well defined time frame, called 'deadline' (Axelsson, 1997a; Koopman, 1996; Soklic, 2002; Edwards et al, 1997; Grehan et al, 1998; Wirth, 2001; Berger, 2002; Gupta, 2003; Karzai et al, 2003; Wilmshurst, 2001; Vahid and Givargis, 2002).
- **Hard real time:** the timing deadline can not be missed; this could produce catastrophic results to the environment the ES controls (Axelsson, 1997a; Yen Yen and Wolf, 1996; Gupta, 2003)
- **Soft real time:** some small delay on the meeting the timing deadline could be acceptable (Axelsson, 1997a; Gupta, 2003)

Kopetz (1997) provides a comparison between hard and soft real time systems:

Table 1.2: Hard real-time versus Soft real-time

Characteristic	Hard real time	Soft real time
Response time	Hard (required)	Soft (desired)
Peak-load performance	Predictable	Degraded
Control of pace	Environment	Computer
Safety	Often critical	Non critical
Size of data files	Small/medium	Large
Redundancy type	Active	Checkpoint (recovery)
Data integrity	Short term	Long term
Error detection	Autonomous	User Assisted

(ii) Reactive vs Interactive:

- **Reactive:** the ES interacts with the environment on the environment's speed (Lee, 1997; Lee, 2000; Gupta, 2003; Vahid and Givargis, 2002). The interact is continuous, which means that inputs from the sensors come in random time intervals and the ES has to react to them (Yen Yen and Wolf, 1996)
- **Interactive:** the ES interacts with the environment but it does it in its own speed (Lee, 1997; Harel et al, 1990).

(iii) Event Triggered vs Time Triggered (Kopetz, 1997; Gupta, 2003):

- **Event Triggered:** the ES responds only whenever an event (a change on the state of its environment) occurs
- **Time Triggered:** the ES responds to its environment in predetermined time intervals

(iv) According to response in case of failure (Kopetz, 1997; Gupta, 2003):

- **Fail Safe:** in that kind of ES, when a failure occurs, all the activities are stopped, so any further damage could be avoided. However, this is a system and not an ES characteristic
- **Fail Operational:** even in the case the ES fails, some basic (minimum) level of service is provided, so any further damage could be avoided

- **Guaranteed Response:** whatever the circumstances, there will always be a response
- **Resource Adequate:** this consists the basis for the 'guaranteed response' ES: it assumes that the ES will have enough resources to provide the 'guaranteed response'
- **Resource Inadequate:** the opposite of the resource adequate
- **Best Effort:** The ES will do 'its best' to provide a response.

Today, embedded systems account for the 22% of the car, and it is predicted that until the end of 2010 this percentage will rise to 35% (Edwards, 2003). As cars comprise higher and higher degree of automation, the percentage of embedded systems increases and the need for controlling their cost has become of vital importance for the automotive companies, especially due to the volume effect. Until recently, automotive OEM developers in order to minimize overall embedded systems costs were concentrating on reducing production costs (Bouyssounouse and Sifakis, 2005, Debardelaben et al, 1997). However, nowadays it is the D&D cost that rises (Bouyssounouse and Sifakis, 2005).

Development cost is defined by Stewart (1995) as ''*the cost of a system up to the point where a decision is made to produce an initial increment of the production units or the operational system''*. The design and development of embedded systems is driven by the conflicting demands of superior functionality, high dependability and minimal cost (Kopetz, 1995) and is a function of parameters that are very difficult to be quantified, like the ability or the experience of the designers involved, and it is therefore very difficult to provide metrics for this cost (Axelsson, 1997a; Roy et al, 1999a; Roy et al, 1999b; Walkerden and Jeffery, 1996). These parameters are appearing as global constraints that affect the design decisions (Stipanovits and Karzai, 2001; Voros et al, 2003).

This thesis is concerned with ES D&D cost estimating on the early stage of the product specification within the automotive industry, and its focus is on producing and delivering a model to estimate the cost of ES D&D in that stage.

1.2. Electronics Cost Estimating Within Automotive Industry

70% to 80% of the product cost is committed at the conceptual phase (Roy et al, 1999; Cokins, 1998; Heikkinnen, 1997). Debardelaben et al (1997) note that although the front-end design process typically involves less than 10% of the total prototyping time and cost, it accounts for more than 80% of a system's life cycle cost. So, it is apparent that the most opportunities for cost savings are in the initial system specification and design phase.

Estimating the D&D cost of an ES accurately is a very difficult task, because they consist of both HW and SW, and therefore any cost model should comprise these two costs (Koopman, 1996), with the integration of SW and HW being one of the main sources of cost in the D&D of ES (Sgroi et al, 1999) and therefore any estimating model should take into account all these costs.

Nowadays, all automotive OEM's outsource the majority of parts that make a car, including electronic parts. This is due to savings in time and resource consumption over in-house production. As the time, machinery, knowledge and skill resources of the company are not enough to produce a good quality product quickly, it is faster, cheaper and easier to outsource those parts to specialised companies who have better resources and more knowledge to do it.

So, OEMs write the system's specifications which are then passed to their suppliers, who are then responsible to provide the requested item with the indented functionality. However, as outsourcing increases, the product data that the Original Equipment Manufacturer (OEM) holds decreases due to the minimal involvement in the actual technology used to design, develop and manufacture the product, which also restricts OEM's access to product cost data as well.

To overcome this issue of not having access to the necessary information, industry is heavily depended on the knowledge of people who have been involved in ES D&D projects in the past in order for them to provide an estimate (expert judgement). In case data from ES D&D past projects are available, then the judgement of experts is supported by these data. However, it is very rare for industry to hold data from past projects, so, expert judgement based on past experience is the approach followed by industry.

The problem with this approach is that the estimator has no access to necessary information for performing his estimation, because information like the effort for writing the SW code or developing the ASICs (application specific integrated circuits) are protected by supplier's IP rights and they are not disclosed to him, so he has to estimate based on the best of his knowledge. In addition, this information is not available on the specification stage.

Models for predicting the effort for D&D an embedded system have been proposed by academia. The common characteristic of these models (they are presented at chapter 2) is that they separately estimate a cost for SW (using Lines of Code or Function Points and the COCOMO model) and a cost for HW (summing up the HW components) and then they use a combination to estimate the complete embedded system cost. However, not all the prices of the HW components can be obtained (for example, the price for an ASIC can not be obtained since it is protected by supplier's IP) and in most of the cases there is no access to the SW code for the Lines of Code or Function Points to be estimated and inserted on the COCOMO model). Another point is that as in industry, all this information is not available on the specification stage.

1.3. Research Context

The author's thesis is concerned with developing an estimation model for assessing the cost of designing and developing an electronic item during the specification stage. This will help on making the estimation procedure structured and consistent, easy to be used and reused. To achieve this, a cost estimating model is developed, in order to provide a formalised framework for the cost estimation. The motivations for industry to adopt this novel approach are summarised as:

- Explicitly represented rationale can help individual estimators clarify their thinking about the generation of an estimate
- Assumptions and exclusions are reduced
- The use of expert judgement becomes a more structured, consistent process;
- The estimation process becomes structured and consistent, available for reuse by experts and non-experts

1.4. Industrial Collaborator: Ford Motor Company

This research is a collaboration between Cranfield University and Ford Motor Co. Ford is one of the leading car manufacturers in the world. It is a multinational company, with its headquarters located in Detroit, USA. Over recent years the Company has been increasing its brand image by making acquisitions, building the Premier Automotive Group (PAG) under the Trust-mark of Ford Motor Company. PAG acquisitions include Jaguar, Aston Martin, Volvo, and most recently Land Rover. Its automotive-related services include Ford Credit, Hertz and Quality Care. The Trust-mark suite is completed with Mazda, Lincoln and Mercury.

Today, Ford Motor Company is the world's largest producer of trucks and the second-largest producer of cars. The company has operations in more than 30 countries, and employs more than 340,000 men and women at its factories, laboratories and offices around the world. Additionally, about 60,000 companies worldwide supply Ford Motor Company with goods and services. The company's annual sales exceed the gross national products of many industrialized nations. In 1998, Ford Motor Company sold more than 6.8 million vehicles worldwide. Ford Motor Company is ranked second on the Fortune 500 list of the largest U.S. industrial corporations, based on sales. In 1998, worldwide sales and revenues totalled $142.6 billion. Net income, excluding one-time items, was $6.5 billion. Although Ford Motor Company is best known as a manufacturer of cars and trucks; it produces other products, including industrial engines, glass, plastics, and a wide range of automotive components. Ford also is established in many other businesses-including financial services, automotive replacement parts, and electronics.

Ford Motor Company places great emphasis on cost control, particularly in the area of multi-billion dollar annual expenditure on externally purchased components. To enable Ford to attain ‘World’s leading consumer company for automotive products and services’ position, it is essential that globalised sourcing is placed with competitive piece prices. To ensure that this is the case, Ford maintains a significant workforce tasked specifically with the control of prices paid for components from outside vendors. One of the main control methods is through the development of detail estimates, which are used as identifiers of uncompetitive pricing and subsequently for price vs. estimate negotiations.

Ford Motor Co., realising the importance of developing a procedure of controlling the cost of electronic parts, initiated the E-Mode project in conjunction with Cranfield University. Through the E-Mode project, Ford Motor Co. wishes to develop a deeper understanding of benchmarks, cost breakdown and cost/technology trends on electronic modules - in particular relating to the impact of volume increases, technology steps that are leading to significant improvements in the function/cost ratio, and the impact of development and software costs to the piece price. The outcome of the proposed research will make the cost estimation process for the Electronic Parts more transparent and realistic.

1.5. Cost Estimating at Ford Motor Company

Within Ford, Cost Estimating is a unique and specialised department within the Finance Organisation. The prime role of Cost Estimating is the estimation of variable and tooling costs for all externally purchased parts. Encompassing forward model programs and presently produced components, Cost Estimating provides direction and support with the aim of ensuring that 'value for money' is achieved through vendor costs. Cost Estimating services many customers including Purchasing, Product Engineering, Program Offices and Plant Vehicle Teams.

Presently aligned with the TVM (Team Value Management) Cluster organisation, the Cost Estimating department operates within a matrix organisation, facing off to both functional (vertically) and operational (horizontally) management. Internally, Cost Estimators have both commodity and vehicle line or powertrain responsibilities to mirror the structure of the department as a whole.

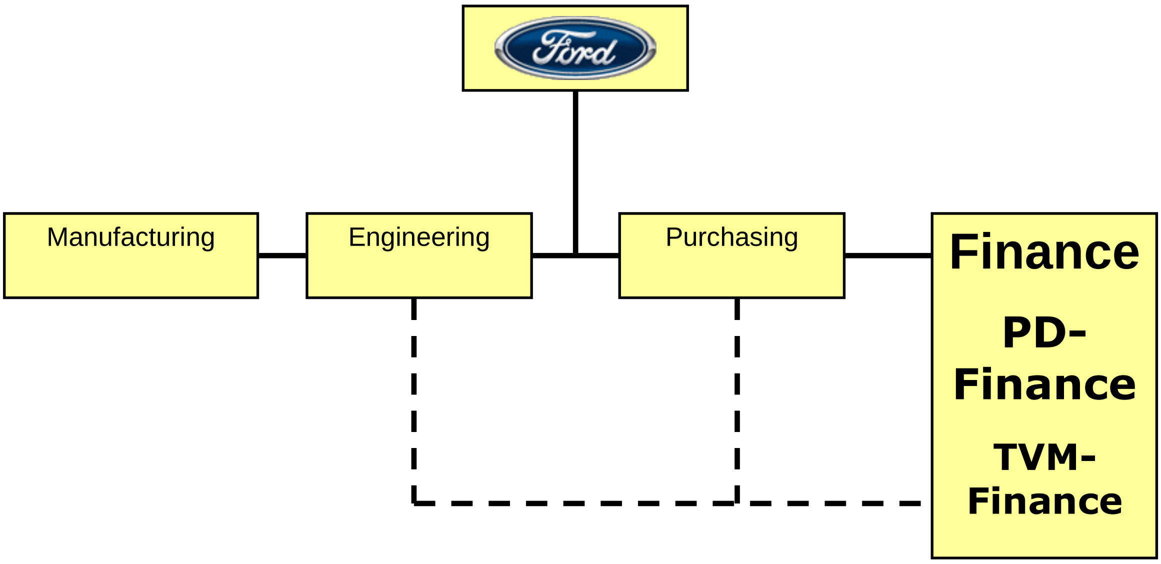


Figure 1.3: Cost Estimating within Ford

In Europe, Ford Cost Estimators are located in Basildon, UK (38), Cologne, Germany (58) and Valencia, Spain (3). In addition, North American Cost Optimisers are located in Dearborn, Michigan. Ford Cost Estimators have close contact with others in wholly owned subsidiaries, Jaguar, Volvo, Mazda and Land Rover.

Electronic Parts Cost Estimating is performed by the Electrical Group and it is staffed with 9 people. The Electrical estimating group is situated under Finance in Ford's organisational structure, and it supports Finance (what if scenarios, future reference), Engineering (late changes, part changes), Purchasing (choosing best price) and Team Value Management (TVM).

1.6. Research Aim

Based on what has been presented so far, the aim of this research is:

''to formalise the cost estimating process for the Design and Development of embedded systems in the specifications stage in the automotive industry''.

Providing a framework to formalise the ES D&D cost estimation process is the major focus of this research. This will offer estimators and/or engineers a structured way of performing ES D&D process at the early stage of product specifications and therefore help them controlling ES costs and on performing system's price evaluation and price negotiation.

1.7. Thesis Structure

Figure 1-4 illustrates the thesis structure, which outlines the approach used to achieve the thesis aim. In Chapter 2, a structured account of the literature is presented, analysed and evaluated. Firstly, the evolution of the ES D&D from independent (for SW and HW) design requirements to today's Platform Based Design is presented, whereas, in a second stage, a review of the ES in automotive and other industries is performed. After that, a view on the specification approaches within the automotive industry (the way specifications are expressed and communicated) is presented, whereas in the next stage an analysis of the ES Cost Estimation and in particular of the ES D&D Cost Estimation is apposed. Finally, the Gap Analysis and the Conclusions along with the Key Observations concludes the chapter.

In Chapter 3, an introduction to the thesis research methodology and objectives, resulted from the literature critical analysis, follows. To accomplish the research objectives, the researcher analyses and evaluates different alternatives in order to design the most suitable research methodology for addressing the research objectives.

Chapter 4 describes the AS-IS study. This is the initial data collection performed using semi-structured questionnaires, interviews and workshops. This study captured the current situation regarding embedded systems D&D Cost Estimation in automotive and aerospace industries. It also presented the commonalties and differences between them two and concluded in identifying the need for a structured D&D cost estimation process.

These conclusions, along with the findings from the literature review, lead to Chapters 5, 6 and 7, which describe the main research contributions and how each of the research objectives is met. In chapter 5, a workshop that helped identifying in detail the problems automotive industry faces when estimating ES D&D cost is initially presented. After that, based on the outcomes of this workshop, the results of the AS-IS study and the literature findings, the researcher proceeds on creating the first part, the *ES HW D&D effort estimation module*, of the overall ES D&D effort estimation framework. In chapters 6 and 7, the researcher presents the development of the rest parts of the ES D&D effort estimation framework, the *ES SW D&D effort estimation module* and the *ES Reuse and Integration D&D effort estimation module.*

The researcher presents the overall ES D&D effort estimation framework in chapter 8. Initially, a presentation of the complete framework in an Excel-based implementation and the way it works is presented. In the next stage, the framework is validated by 3 different automotive OEMs. The framework represents a generic method that integrates the functions and data identified throughout this research to a support a structured cost estimate.

Chapters 8 synthesises the work of the thesis by discussing the findings, the opportunities and the implications of the research outcomes. Areas for future research are discussed before concluding the research. The conclusions respond to the stated aim and objectives of the thesis.

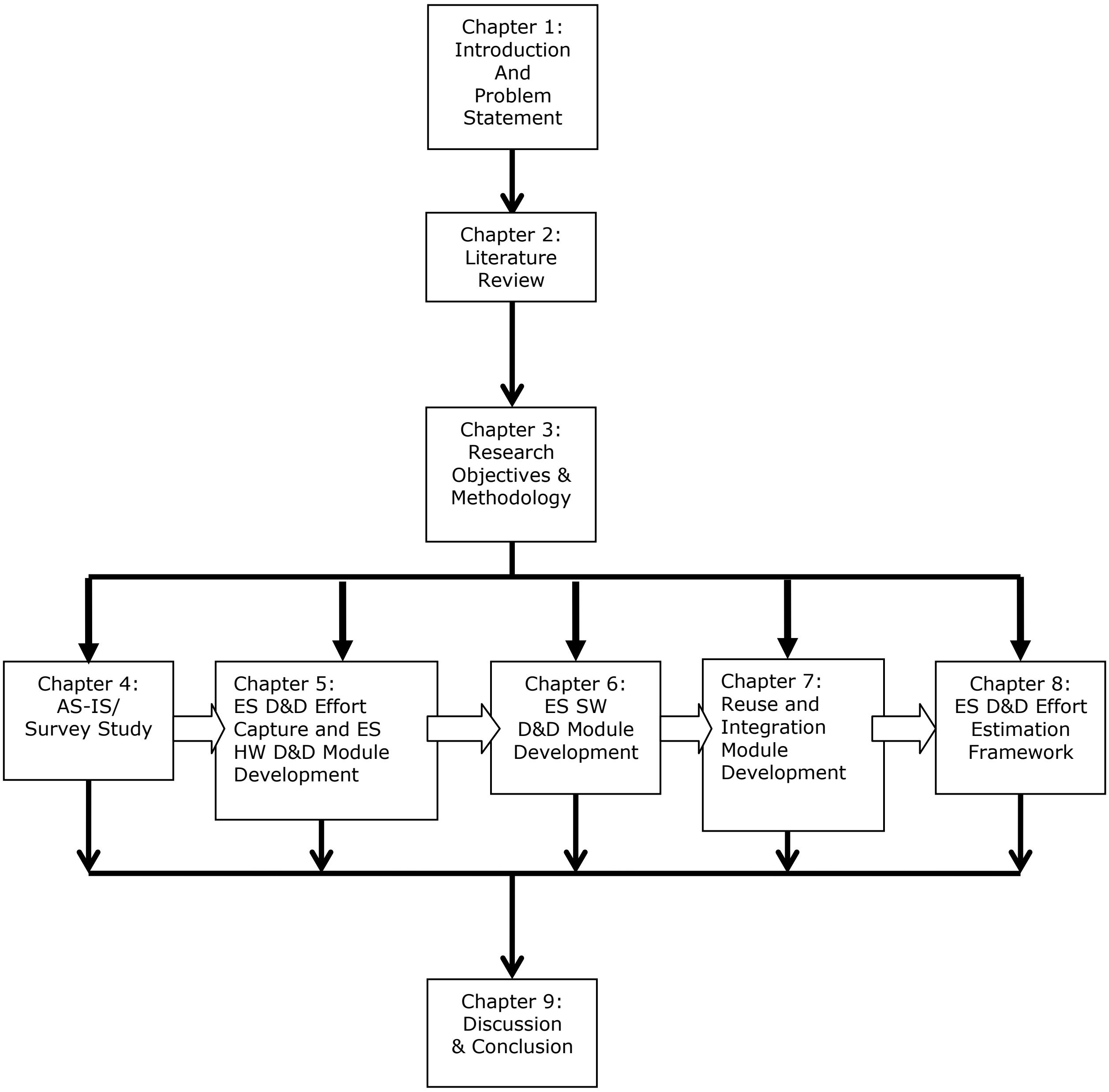


Figure 1.4: Thesis structure

2. Literature Review

The previous chapter presented a general view of the research domain and outlined the problem this research is aiming to address. In this Chapter a structured account from literature is being presented, providing background information to support the view that there is a need for the development of a process/method/approach/framework for electronic items cost estimation based on the item's specifications.

This chapter is divided into several sections. Section 2.1 explains the ES D&D domain and presents ES D&D evolution through time. The current status in ES D&D is presented in section 2.2 for the automotive industry and in section 2.3 for the other industries. Section 2.4 presents the alternative forms of ES specifications, whereas section 2.5 presents various approaches on ES Cost Estimation. Estimating ES D&D cost is presented in section 2.6, whereas gap analysis and chapter summary and key observations follow in sections 2.7 and 2.8 respectively.

2.1. Embedded Systems Design and Development

The design and development of embedded systems is a very difficult problem, because of (Axelsson, 1999; Sciuto, 2000; Lee, 2000; Sztipanovits and Karsai, 2001; Voros et al, 2003):

- The alternatives available,
- The problems in embedded software development and its integration with the hardware,
- Because it requires people with specific knowledge of the environment the embedded system will be operating within,
- HW and SW are being developed independently by people who either understand only the SW part or only the HW part of the embedded system; there are not many people who understand the complete system, and
- Because various parameters of the design process can not be quantified and included in equations (company policies, preference to a specific supplier, engineer's experience, etc), which they appear as global constraints and affect the design decisions.

In addition, as it was described earlier:

- ES consist of both HW and SW who have to work seamlessly in order for the ES to fulfil its task,
- They are created to address a specific situation, so they must be designed for that particular task, and
- They have to comprise with all the limitations imposed on them.

Under these findings, the design of an embedded system is a complicated task, as the designer has to trade off between alternative implementations of the ES and choose the one that optimises most of the factors that affect his design (Vahid and Givargis, 2002; Axelsson, 1997a; Lee, 2000; Sciuto 2000). These competing factors that affect the design are called design metrics (Vahid and Givargis, 2002; Axelsson, 1997a): (i) NRE (non recurring) cost, (ii) Unit cost (without the NRE cost), (iii) Size, (iv) Performance, (v) Power Consumption, (vi) Flexibility (easiness on changing system's functionality), (vii) Time to create the first prototype, (viii) Time to market, (ix) Maintainability, (x) Correctness, (xi) Safety, (xii) Production Cost and (xiii) Product Complexity. This process of evaluating alternative implementations until one of them is chosen and implemented, is called '’Design Space Exploration'’ (Vahid and Givargis, 2002; Kalavade end Lee, 1993; Hsieh et al, 2000) and it is described by Hsieh et al (2000) as '’the process of analysing several 'correct' implementation alternatives to determine the most suitable one'’.

Making a change to one of these factors (ie complexity of the product) will have an affect in another factor (ie more time to market, or more time to create the first prototype, or bigger NRE cost, etc). The designer, in order to choose the optimum solution between all these different potential alternatives, is based on his experience (Berger, 2002) and for that reason the designer must be competent in both SW and HW areas (Vahid and Givargis, 2002; Wirth, 2001).

2.1.1: Traditional Design

Until recently, when an ES was designed, it was separated in its HW and SW subsystems from the beginning of its design. The two parts were designed separately and they were brought up together only very close to the end of the system's design process:

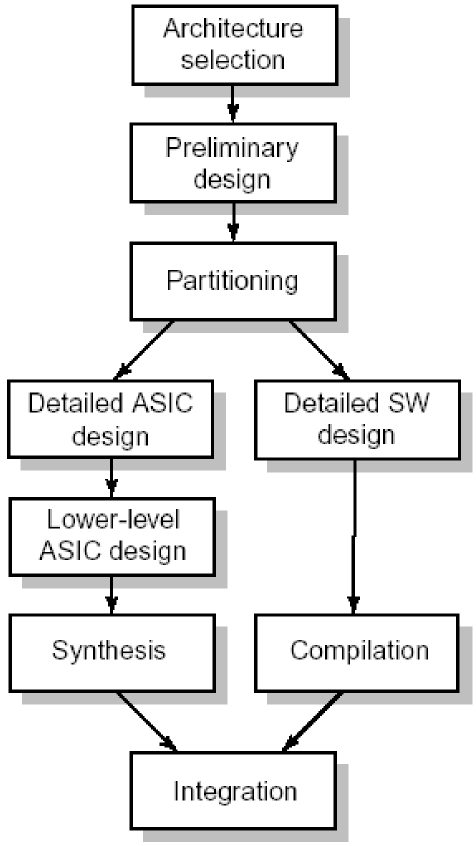


Figure 2.1: Traditional Design Flow (Axelsson, 1997a)

Axelsson (1997a) notes that adopters of this design process were assuming that an implementation can be realised by adopting a predetermined architecture and by implementing functionalities on it. However, as many authors point out (Edwards et al, 1997; Gajski et al, 1997; Gajski et al, 1998; Lavagno et al, 1998; Domer et al, 1998; Sangiovanni-Vincentelli, 2000; Axelsson, 1999; Axelsson, 1997a), this approach limits seriously the ‘’design space’’, since potential optimum solutions are excluded, and problems are very likely to appear on the later stages of the design process. Following this design approach, the final system was a result of the designer’s experience and expertiece, not a decision that comes out of a methodological approach of evaluating all potential alternatives (Gasjki et al, 1994; Edwards et al, 1997).

The second major problem of this design process is that embedded software was produced independently from HW and in most of the cases only after hardware had already been developed (Lavagno et al, 1998). This created additional problems on the integration stage at the end of the design process (Axelsson, 1999; Berger, 2002; Ernst, 1998). Concluding, in this design process errors were not disclosed until a very late stage, making their correction very costly, because then serious rework had to be applied which would also result in delays to the project's time schedule (Debardelaben et al, 1997). As (Axelsson, 1997b) notes, ''problems are created early and discovered late''.

2.1.2: Co - Design

To overcome these problems, and to reduce the time to market and development cost, researchers moved towards the HW/SW co-design (co-design thereafter) approach. In co-design, the designer separates HW and SW only after he has evaluated and benchmarked the available implementations based on the design metrics (Kalavade and Lee, 1993). In this approach, HW and SW are being designed and developed in parallel and concurrently, allowing that way better design space exploration and optimisation by trading off functions between SW and HW at any time during the design process, resulting to less HW/SW integration problems as well (Axelsson, 1999; Vahid and Givargis, 2002; Kalavade and Lee, 1993; Sangiovanni-Vincentelli, 1998; Axelsson, 1997a,b). In addition, by adopting the co-design approach, changes can be identified early, avoiding costly revisions in the latest stages of the project. Figure 2.2 shows the shorter development time (and therefore shorter time to market) that can be achieved using the co-design approach:

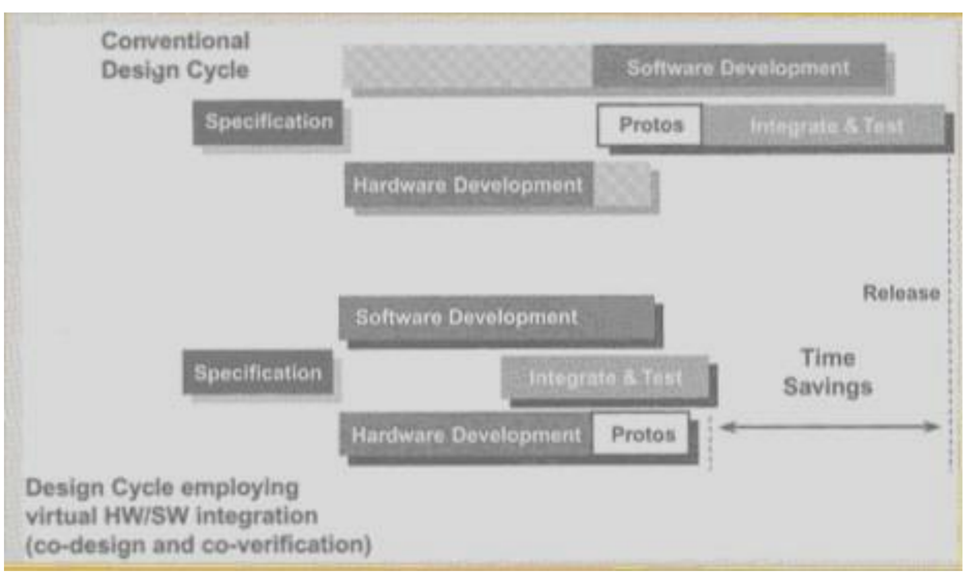


Figure 2.2: shorter time to market using co-design (Berger, 2002)

Co-design is an active research field that has experienced various developments during the recent years. However, even today in industry software and hardware are developed concurrently but independently, with the two development teams communicating through informal specifications (Voros et al, 2003). The reason for SW and HW still being developed independently is that various parameters of the design process can not be quantified and included in equations, so, they are appearing as global constraints that affect the design decisions (Sztipanovits and Karsai, 2001; Voros et al, 2003). Examples of these parameters are environmental issues, engineer's expertiece with a particular technique or processor model, company policy that dictates the components to be used, etc. Berger (2002) shows the close commonalties in the activities performed in the partitioning stage from both the HW and SW teams:

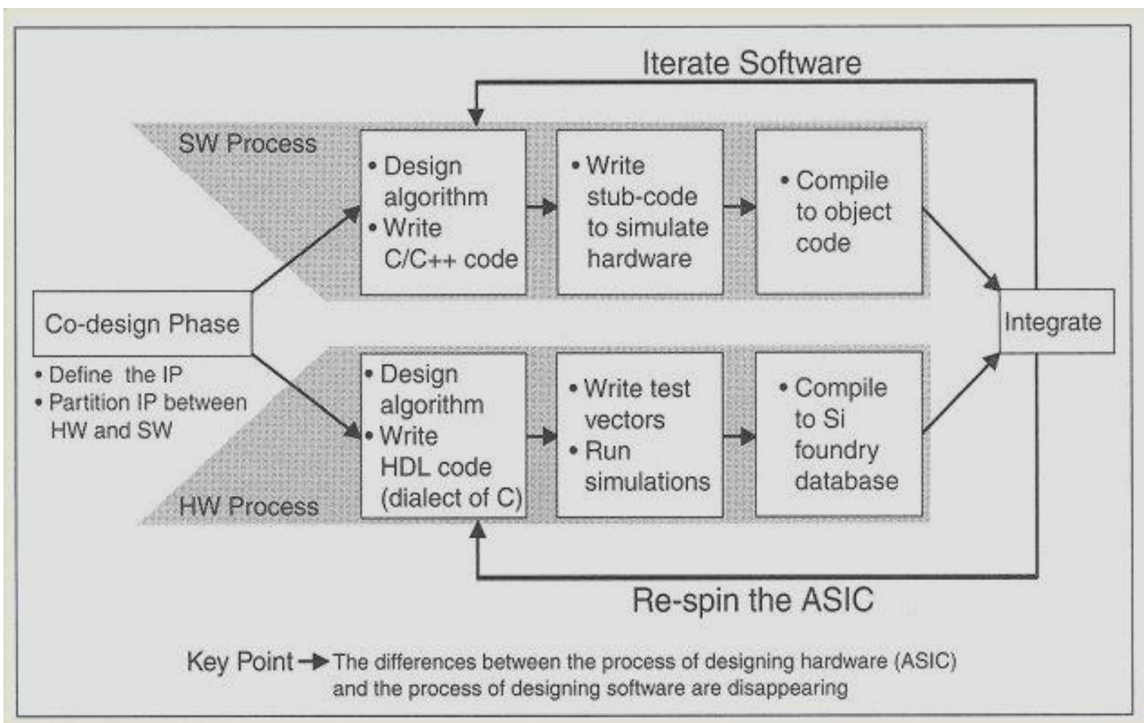


Figure 2.3: Commonalties in SW and HW development teams (Berger, 2002)

Sangiovanni-Vincentelli (2000) and Lavagno et al (1998), note that co-design is used very late in the design process, after important decisions on implementation have been made. They argue that unless there is a formal description of the system's functionality expressed without any ambiguity and with no biased preference towards a specific solution (ie due to the engineer's expertise or experience), the engineer could be designing a solution that is not the optimum one. They suggest that the design process should start from the system level, describing the system's functionality using mathematical models (Lee, 2000; Lee, 2002; Edwards et al, 1997).

Using this approach, the specifications of the system are described without any ambiguity, and the co-design of the system follows a well established process, where its final implementation comes as the optimum solution between various alternatives (Gasjki et al, 1994; Edwards et al, 1997). Therefore, in the recent years co-design has moved towards system-level co-design, since the issue is not only partitioning functionalities between SW and HW, but co-designing an ''optimum'' implementation derived by evaluating various alternatives and by trading off between the design metrics (Axelsson, 1997a).

2.1.3: System-level design and model-based design

System level design starts with the engineer creating a formal model of the system to be designed based on the system's specification provided. The designer creates a conceptual model of the target ES by dividing the target ES in subsystems, even individual items, and by also noting how these items and subsystems should work together, achieving the desired overall system functionality (Gajski et al, 1997). This functionality should be described in so much detail that misunderstandings or misinterpretations are eliminated, because these misconceptions lead to unnecessary and costly repetitive loops of the design process (Edwards et al, 1997; Gajski et al, 1997; Lavagno et al, 1998; Gajski et al, 1998; Axelsson, 1997a).

The design process, based on the system level design, contains all the potential alternative implementations, and its aim is in each step to remove choices from the original model (Sangiovanni-Vincentelli, 2000). This way, target implementation is the result of a well described step by step design process that is validated and simulated in each stage, ensuring that the final implementation is *'correct by construction'* (Lavagno and Sangiovanni-Vincentelli, 1998), after *''a thorough exploration of all the possible architectural and technological alternatives''* (Gajski, 1994). According to Edwards et al (1997) and Gajski et al (1997) for the model to serve its purpose, it must have the following attributes and satisfy the following characteristics:

- It must be described in such detail that no misconception or misinterpretation is possible
- It should be easy to understood and modified
- It should contain the properties the ES should satisfy
- It should contain a set of indices, in order for the design metrics to be benchmarked against
- It should also contain any set of constraints imposed either in the design itself or in the ES properties

To achieve this, the behaviour of the sub-systems and the behaviour of the system itself is captured using mathematical models, also known as ‘‘Models of Computation’’ (MoCs). The behavioural description specifies the indented functionality of the system's actual implementation and what the performance requirements are (Axelsson, 1997a). MoCs use mathematical descriptions and a set of syntax and rules to describe and compute the behaviour of the model, subsystem or individual item and the relationships between them (Edwards et al, 1997; Lavagno et al, 1998; Gajski et al, 1994). Various MoCs have been proposed (Gajski et al, 1997; Edwards et al, 1997; Lee and Sangiovanni-Vincentelli, 1998), however the most important ones are Finite State Machines (FSM), Data Flow and Discrete Event (Lee and Sangiovanni-Vincentelli, 1998; Sangiovanni-Vincentelli, 2000).

Creating a model of a system does not necessarily employ only one model of computation. As the engineer goes through the design process he might use a combination of models depending on which of the system's characteristics he is interested on exploring in this particular stage (Gajski et al, 1997; Lavagno et al, 1998). Nevertheless, this mix of models imposes a major problem on the design process, since transition (semantics and communication rules) between models is not yet well defined (Edwards et al, 1997). Although some approaches have been suggested to address this problem, computational models have significant differences between them and therefore mappings between them have not been available yet (Gajski et al, 1997; Lee and Sangiovanni-Vincentelli, 1998).

System-level design is an active research field that has experienced various developments during the recent years. There have been various system-level co-design frameworks/tools developed, both by academia and industry (Chinook (Chou et al, 1995), Cosmos (Ismail et al, 1994), Cosyma (Ernst et al, 1996), CoWare (Rompaey et al, 1996), Lycos (Madsen et al, 1997), Polis (Balarin et al, 1997), Ptolemy (Hylands et al, 2003), Tosca (Balboni et al, 1996), Metropolis (Balarin et al, 2003; Balarin et al, 2002), Giotto (Henzinger et a, 2001), Artemis (Pimentel et al, 2001), MCSE (Calvez, 1993)). They all follow the same underlying design logic as it is presented bellow:

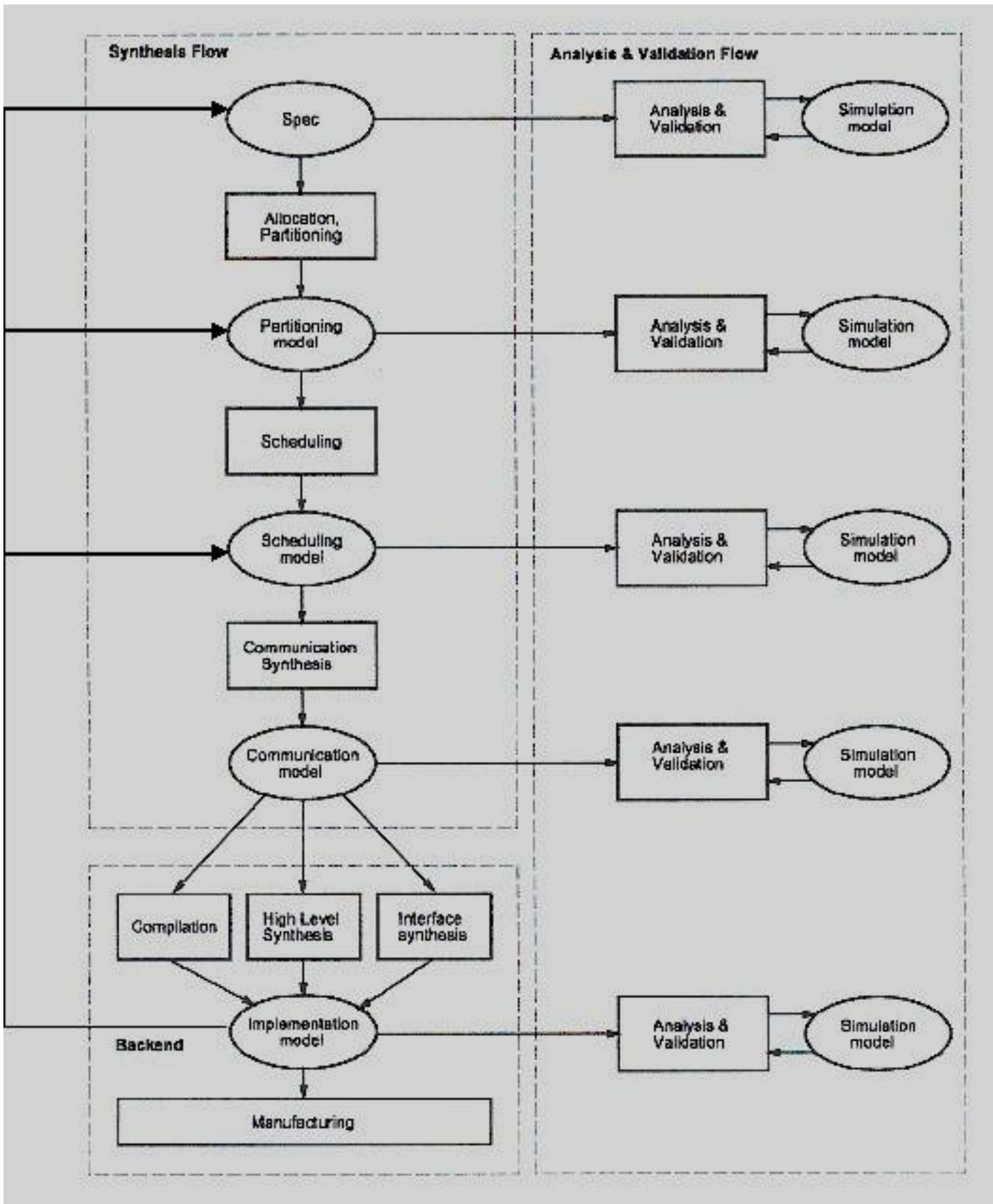


Figure 2.4: model-based co-design flow (Gajski et al, 1997)

Step 1: Extracting an executable specification

After the designer decides which computational model(s) he is going to use, he can then describe the system's behaviour on an executable form (mathematical expressions) using a certain language (ie Ada, VHDL, C, Esterel, Verilog, Matlab, Statecharts, etc) (Edwards et al, 1997; Gajski et al, 1998; Slomka et al, 2000 Sangiovanni-Vincentelli, 2000). This way, the system's behaviour is captured as independently as possible of the target implementation and the distribution of functionalities on SW or HW (Axelsson, 1997a). In general, systems are described as a set of communicating and concurrent entities activated when an event occurs (Ernst, 1998). This 'executable specification' can be simulated, giving feedback to the designer on how correct his design is and also serves as input to the subsequent stages of the design process (Gajski et al, 1997).

Step 2: Allocation

After the specification phase has been completed, the allocation phase follows. In this phase, the designer chooses, most of the times manually, what components and how many of them he is going to use on the system's implementation (Gajski et al, 1997; Axelsson, 1995; Wolf, 1994). The conceptualisation of the architecture is one of the major issues on designing ES (Edwards et al, 1997; Gajski et al, 1998). In general terms, an ES architecture consists of a combination of (i) HW (microprocessors/microcontrollers, ASICs, FPGAs, etc), (ii) SW (RTOS, device drivers, applications, etc), and (iii) means for connecting these items (busses, channels, etc) (Edwards et al, 1997). This way, a conceptual system architecture is formed and the design space exploration process is initialised (Gajski et al, 1997).

Step 3: Partitioning

The next stage on the design process is the partitioning stage, where the designer decides which tasks are going to be implemented on what processing elements of the ES's architecture that was derived in the previous step (Axelsson 1995; Axelsson 1997; Gajski et al, 1997; Gajski et al, 1998; Thiry et Claesen, 1996). Partitioning is done in two steps. In the beginning, the designer evaluates, making quick estimates, trade-offs of moving a task from SW to HW or vice versa, and in the second stage the architecture derived from the first step is refined. Although there have been a number of partitioning methodologies proposed, none has emerged as the de facto winner because of the great complexity heterogeneous ES bring along (Edwards et al, 1997; Gajski et al, 1994).

Step 4: Scheduling

Next in the design process comes the scheduling phase. Scheduling is the step where the designer decides the sequence of the incoming tasks as they will be executed on the respective ES processor(s) (custom processors, ASICs, etc) (Edwards et al, 1997; Gajski et al, 1997; De Micheli and Gupta, 1997). Since ES monitor and interact with their environment, stimuli from the environment will be arriving in random time intervals. So, there must be a mechanism that will assign execution priorities to incoming and exiting tasks and will decide how much of the available time and resources these tasks will each one occupy (Sgroi et al, 1999; Thiry and Claesen, 1996; Engels and Devadas, 2000; De Micheli and Gupta, 1997; Axelsson 1997). This mechanism should even disturb the sequence if for example a task with a higher priority arrives (Edwards et al, 1997; Gajski et al, 1997; Engels and Devadas, 2000).

Step 5: Communication Synthesis

After all these stages have been completed, then the communications synthesis is performed. In the scheduling stage, the designer defined how the sequence of actions imposed on each of the architecture items is going to be performed. In this stage, the communications stage, the designer decides on how interactive tasks should work together for the desired functionality of the ES to be realized (Gajski et al, 1997; Ortega and Borriello, 1997).

Step 6: Synthesis

In the previous stages, the designer derived both the ES's architecture – in an abstract level – and how each of this architecture's items should perform. In this last stage (shown in Figure 2.4 as 'Backend'), the designer using specialised tools will derive the actual implementation of the architecture's parts and therefore of the ES itself (Gajski et al, 1997; Slomka et al, 2000).

The synthesis of the HW part (creating HW implementations out of the computational model) has been explored and developed by various authors in great extent. In the other hand, SW synthesis is a more recent problem (Edwards et al, 1997), because so far embedded SW was not considered important (Lee, 2002) and most of the implementations were realized on HW (Lavagno et al, 1998). However, nowadays where a considerable shift to SW is being observed (Humphrey, 1992) the need for advanced tools for communication synthesis has become of absolute importance.

Validation and Analysis

Validation is the process of assuring that a design (in every stage) is correct (Edwards et al, 1997; Thiry and Claesen, 1996) and analysis is the process of evaluating some key indices (or metrics) so the designer can make better design decisions. If the designer, in any stage, can not validate or analyse his design, he then has to return to previous stage(s) and re-examine his decision(s) (Gajski et al, 1997). In order to ensure that the less possible iterations will be executed, analysis and validation should be performed in each of the design steps, as the design process proceeds from one stage to another. By this way, the design metrics and the correctness of the design are evaluated, providing valuable information to the designer, assisting him on performing better decision making (Edwards et al, 1997; Gajski et al, 1997; Thiry and Claesen, 1996). This is the reason that analysis and validation are shown as a separate flow alongside the design process in figure 2.4.

Validation can be performed either by simulation or formal verification. Simulation can be performed only for SW (compilers, debuggers, simulators, etc), only for HW (testbenches, fast prototypes, etc) or for HW and SW together (co-simulation, via simulators, testbenches, emulators, etc.). Formal verification is performed either by specification verification, which examines if a property of the design is satisfied, and implementation verification, that checks if the implementation derived by a computational model satisfies any property or constraint imposed in this specific implementation (Edwards et al, 1997; Gajski et al, 1997; Olukotun et al, 1998). CoWare (Rompaey et al, 1995) and Seamless (Klein,

1996) are two of the most used in industry tools for co-simulation and co-verification (Voros et al, 2003).

Analysis is the procedure of evaluating some key indices (or metrics) so as the designer can make better design decisions (Gajski et al, 1997). The analysis of each design (in each stage) can be done using a static analyser, a profiler or a visualiser. The static analyser associates each item's functionality with some of the design metrics and then uses probabilistic techniques to derive an estimation of these metrics. The profiler is a useful tool for describing dynamic situations, like for example branch execution time, whereas the visualiser is used to visually describe the structure of the design. It can also show the relationship of data between different design views in a synchronised view (Gajski et al, 1997).

2.1.4: General observations on model based design

Voros et al (2003) present a survey of the industrial practice regarding embedded systems design. Their findings agree with the procedure presented above. They note that the design starts with an informal specification, which describes the system in a very abstract way, and which the functional specification comes from, using the appropriate computational model. They agree that different models of computation are used throughout the design process, depending on the system features that are of particular interest for the engineer at the specific time, and that partitioning and architecture exploration are heavily based on the designers' experience. They also note that in the synthesis level, HW development (performing simulation based on High Description Languages (HDLs) and synthesis using commercial tools) and SW development (the SW is described in a HDL like C and then compilers are used to derive machine code) are being developed independently but concurrently, with the SW and HW teams to communicate with informal specifications, in order to arrive on creating the actual ES.

As it was shown earlier, the first step of the design process is a well defined functional specification (Edwards et al, 1997; Gajski et al, 1997; Lavagno et al, 1998; Gajski et al, 1998). However, most of the times, the specification is not described in the required detail. In addition, the use of different models of computation throughout the design process as well as the different skills and expertiece of the design engineers add up to the co-design process complexity (Edwards et al, 1997; Voros et al, 2003). Due to these reasons, as the designer tries to transform the functional specification to an implementation, he has to perform as much iterations of the design process as required, in order to reach the desired implementation, adding more information as he goes through each step and iteration (Edwards et al, 1997; Gajski et al, 1997; Voros et al, 2003). Iterations may also occur during the analysis and validation stages.

Iterations (and trade offs) do not only occur between the stages of the design process, but within each of the stages as well. For example, within the partitioning stage, a partitioning decision can be altered if the designer decides to change the allocation of a task, or an architecture decision if for example the designer decides to add a new processor or an additional memory chip (Axelsson, 1997a; Voros et al, 2003). For every change the designer makes, he has to check that the new decision satisfies any constraints imposed either to the overall design or to this particular design stage (cost, performance, development time, etc) (Edwards et al, 1997; Gajski et al, 1997; Gajski et al, 1998; Vahid and Givargis, 2002). In practice, the architecture, the scheduling and the partitioning are developed interacting with each other, since any change to each of them influences the others (Axelsson, 1997a) and they are most of the times based on the designer's previous experience (Voros et al, 2003). This is the reason for the backward arrows on Figure 2.4, to show the interactions within the design process stages and that the co-design process is iterative.

Although various system-level co-design frameworks/tools exist, none of them has been established as a winner and they are rarely followed in industry. On the contrary, companies develop their own methodologies, based on their internal policies, experience and expertiece, since various parameters of the design process can not be quantified and included in equations (like environmental issues, engineer's expertiece with a particular technique or processor model, company policy that dictates the components to be used, etc.) which they appear as global constraints affecting the design decisions (Sztipanovits and Karsai, 2001; Voros et al, 2003). In addition, none of these frameworks/tools is covering the whole co-design process as it is presented here but only some parts of it: others cover the synthesis stage (ie Cosyma), others are more powerful in the validation stage (ie Polis), etc (Gajski et al, 1999).

2.1.5: Co – Design with IPs

Nowadays, more and more ES contain IPs (Intellectual Property). IPs are items (either HW or SW) that the companies that manufactured them protect with

Intellectual Property Rights in order to protect the design knowledge behind them and retain their competitive advantage (Gajski et al, 1999; Vahid and Givargis, 2002).

Recognising the more and more extended use of IPs in ES design, (Gajski et al, 1999) derived a methodology (the SpecC design methodology) for incorporating IPs on the model based design methodology and the SpecC language (Gajski et al, 1999; Gajski et al, 1998), which due to its ability of capturing all the computational models used on the design process (it is used by all models in all the stages), it is able to provide a unified view of the system. This way, the insertion or the exclusion of an IP or a component can be easily identified and addressed, since the unified system representation does not allow any ambiguity on the transformation from one model to another. Figure 2.5 shows the SpecC methodology, proposed by (Gajski et al, 1999):

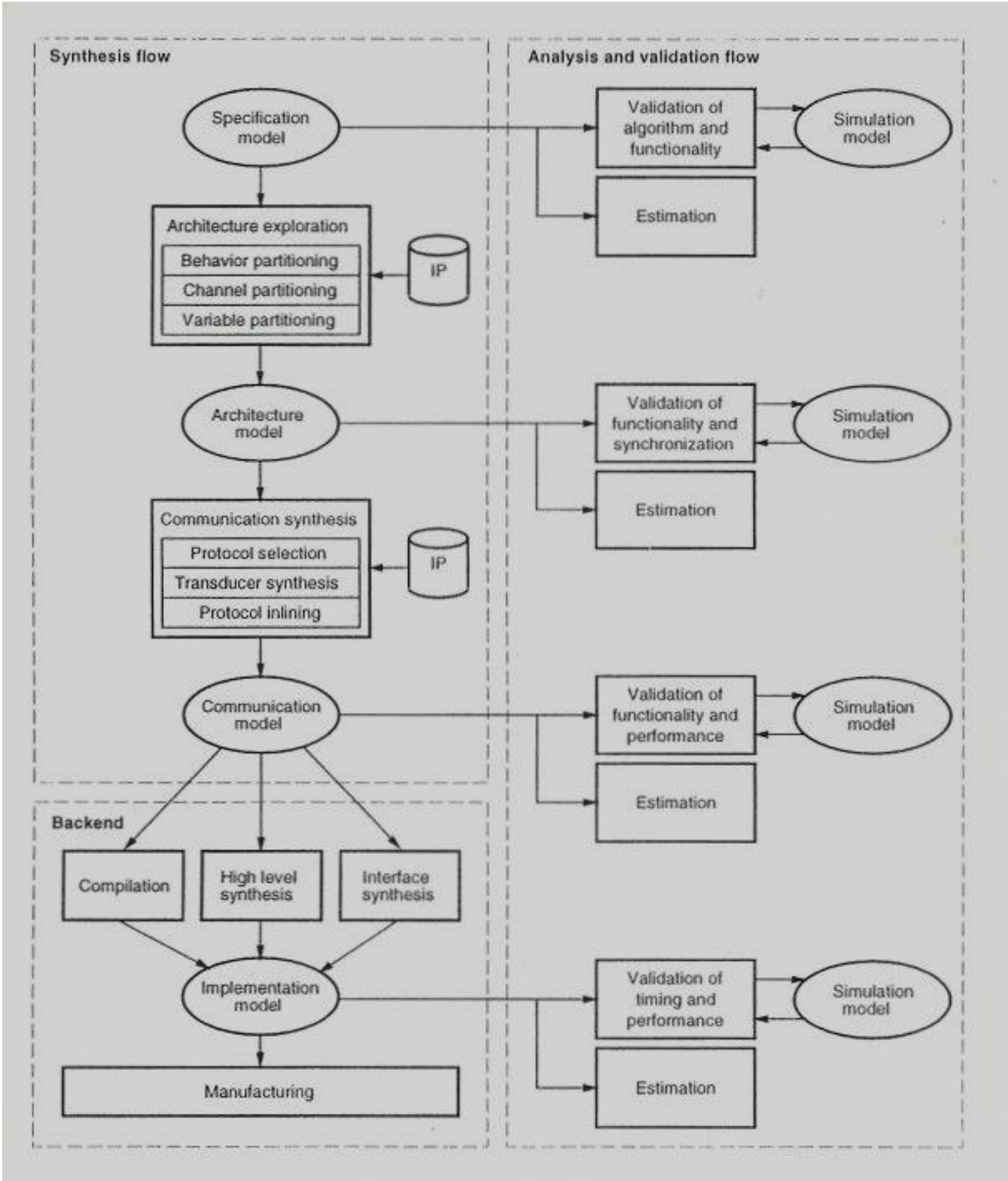


Figure 2.5: Design flow using IPs (Gajski et al, 1999).

2.1.6: Co – Design with Reusable components

In engineering in general, and in electronics engineering in particular, reuse is something more than the standard practice. Circuits, microprocessors and other HW components are too expensive to be developed each time a new electronic item is designed; software artefacts (modules, code, objects, etc) are also being reused, because of their flexibility (DTI, 2004a; DTI, 2004b).

In addition, in industries that are characterised by high production volumes such as automotive, it is a common practise for previously manufactured products to be reused; reuse is considered to be a key for a high level of productivity, shorter product supply times, low manufacturing product costs and a high degree of quality (Schmietendorf et al, 1999).

Although both SW and HW parts are being reused in the design and development of an embedded system, reuse literature is concentrating on SW reuse. SW is the one that has been reused the most and for longest period due to its flexibility; it is considered easy to modify and parts of it or the complete program can be transferred to a new development, avoiding the long HW development and manufacturing cycles (DTI, 2004a). SW has been reused in developing applications since the early years of programming; it is a usual practice to reuse 'old' code to 'new' developments. However, reuse most of the time is being practiced in an ad-hoc way, restricting its benefits of fully being exploited (Kang et al, 1992).

A review of the literature reveals various metrics that have been proposed for estimating SW reuse (Schmietendorf, 1999; Frakes and Terry, 1994; Frakes and Terry, 1996a,b). These are summarised by Schmietendorf (1999) on the following table:

Table 2.1: Reuse Metrics (Schmietendorf et al, 1999)

Metric	Measure-ing type	Calculation	Scale Properties
Amortization (Gaffney)	Calculation	(RCR + RCWR/n –1) * R +1, n – number of expected reuse, R – proportion of reused code in the product	potentially ratio scaled
COCOMO Modification (Balda)	Estimation respectively Measure-ment	LM = $a_iN_i^b$ N_1= KDSI for unique code developed N_2= KDSI for code developed for reuse N_3= KDSI from reused code N_4= KDSI from modified components	potentially ratio scaled
Consumer/Producer Reuse (Chen)	Estimation	Reuse(S,P) = Σ weights S – Systems, P – Repositories, AppReusePerc(S) = Reuse(S) / Size(S) RepReusePerc(S,p) =Reuse(S,p)/Size(p)	ordinal scaled
Cost Benefit (Bollinger)	Estimation	Benefit = withoutReuse – withReuse - ReuseInvestment	potentially ratio scaled
Cost Benefit (Henders.)	Estimation	ROI= (withoutReuse – withReuse)/ producedReuseComponents	potentially ratio scaled
Cost Benefit (Malan)	Estimation	Costs= (withoutReuse – withReuse) - ReuseOverhead (considering a set of products)	potentially ratio scaled
Cost Benefits (Poulin)	Estimation	Σ costs - Σ benefits	potentially ratio scaled
DISA (DoD)	Calculation	Defense Information Systems Agency: - lines of new custom code - lines of new reusable code (code written for reuse by others) - lines of verbatim reused code - lines of adapted reused code	potentially ratio scaled
Generic Reuse (Bieman)	Calculation	Number of generic classes or functions or parameters (for templates)	potentially ratio scaled
RCA (Poulin)	Estimation	ReuseCostAvoidance=Development CostAvoidance+ServiceCostAvoidance	ordinal scaled
RCR (Poulin)	Estimation	Portion of effort that it takes a similar component without modification (black-box reuse)	potentially ratio scaled
RCWR (Poulin)	Estimation	Portion of effort that is take to write a similar reusable component	potentially ratio scaled
Reusable Index (Sodhi)	Valuation	Component: 4 – most reusable, . . ., 1 – least reusable	ordinal scaled
Reusability (Barnard)	Calculation	Calculation-formula based on CBO, DIT, NFC and other OO-metrics	ordinal scaled
Reuse Level (Frakes)	Calculation	(InternalReuseLevel + ExternalReuse Level)/ TotalNumberOfItems	ordinal scaled
Reuse Leverage (Banker)	Calculation	TotalObjectsUsed/ NewObjectsBuilt	ordinal scaled
Reuse Leverage (Poulin)	Estimation	ProductivityWithReuse/ ProductivityWithoutReuse	potentially ratio scaled
Reuse Percent (Poulin)	Estimation	ReusedSoftware/TotalSoftware	ordinal scaled
ROI (Poulin)	Estimation	ReturnOnInvestment = Σ Coeff * Reuse in Phase i	potentially ratio scaled
RVA (Poulin)	Calculation	ReuseValueAdded= (TotalSource Statements + SourceInstructionsReused ByOthers)/ (TotalSourceStatements – ReusedSourceInstruction)	ordinal scaled
Verbatim Reuse (Bieman)	Calculation	Classes (functions) included by use or has relationships	potentially ratio scaled

In addition to the above, there are company-specific approaches on software reuse metrics, as the ones described at Ferri et al (1997) and Morisio et al (2000).

2.1.7: Latest developments in Embedded Systems D&D

In the recent past, an embedded system would be either small or simple, or the composition of almost non-interacting imported and assembled components (Bouyssounouse and Sifakis, 2005). However, nowadays embedded systems are in most of the cases distributed and they need to collaborate in order to fulfil a complete application (Sciuto, 2000; Axelsson, 1999). In addition, there are some cases (ie automotive industry) of having the same functionality distributed in different Electronic Control Units (ECUs) or various different functionalities sharing the same ECU (Axelsson, 2001b; Bouyssounouse and Sifakis, 2005).

In any of the cases presented in the previous paragraph, a distributed ''functional network'' is being formed (Bouyssounouse and Sifakis, 2005). In that functional network, each node can not be studied separately, since the behaviour of each node is closely linked with the behaviour of the nodes it is connected with. In addition, by independently developing parts that will be later networked, it creates many problems that remain undetected until the final integration phase (Axelsson, 1999). As more and more complex functionality is needed, these systems can not be designed as it was done traditionally and a different design approach, which will use the functional network as the basis for each of the network's nodes development is needed (Bouyssounouse and Sifakis, 2005). This implies that a System Level Language (SLL) is needed for the behavioural description and modelling of the embedded system (Axelsson, 2002).

The SLL should be able to capture and model both the functional network and the environment it will operate within, so all the alternative operating scenarios could be examined (Axelsson, 1999). Although a variety of SSLs is available (ANSI C/C++, SystemC, Java, Superlog, etc), UML seems to emerge as a standard for the system level design, due to its rich notation and modelling capabilities that allow system's structure and behaviour to be expressed in various levels of abstraction (Chen et al, 2003). UML contains the necessary elements and extension mechanisms needed to model embedded systems (Axelsson, 2002): it can capture scenarios through Use Cases, performance through its Tagged Attributes, physical resources through Development Diagrams and time, using Classifiers and Tagged Attributes (Chen et al, 2003; Axelsson, 2002). In addition, UML is a platform independent language: it supports views of the system that can be analysed and refined without early introduction of any implementation decision (Bouyssounouse and Sifakis, 2005).

Until recently, most of the implementations were realized on HW (Lavagno et al, 1998), since embedded SW was not considered important (Lee, 2002). However, nowadays a considerable shift to SW is being observed (Humphrey, 1992): Paulin et al (1996; 1997) and Ziebart (1991) estimate that the percentage of an embedded system development effort used for coding is up to 60%, whereas Lavagno et al (1998) estimate it to 70%. An almost 60% percentage was also observed by the researchers through workshop with engineers from the automotive sector and interviews with experts from other sectors (aerospace/defence). UML is the modelling language of the SW engineering domain (Beeck et al, 2003); so it should be used to model a domain like embedded systems which is dominated by the SW development. This, along with the fact that it is a platform independent language are the main reasons UML is gaining more and more acceptance in the embedded systems design area.

However, there are some weaknesses that prevent the full deployment of UML in the embedded systems domain (Beeck et al, 2003; Bouyssounouse and Sifakis, 2005; Axelsson, 2002; Edwards, 2003):

- Not precisely defined semantics: UML is a semi-formal language; its syntax is formal but its semantics are not. This means that the same diagram could be interpreted differently by two different people. There have been proposals for standardising the UML semantics, but they have not yet been incorporated to the official standard.
- Automatic code generation: There are available tools that can automatically generate code from UML diagrams. The drawback with this approach is that there is no formal agreement on the meaning of some of the UML diagrams yet (since the UML semantics are not yet well defined), so each automated generation code tool vendor makes his own decisions. This means that different tools using the same UML diagram as source, they can produce code that functions differently.

Work is being currently carried out by the Object Management Group (OMG) for standardising the UML semantics. The fact that UML is gaining more and more acceptance and becomes the de facto modelling language in the embedded systems domain led OMG to create –and recently standardise- the UML profile for Scedulability, Performance and Time. This profile, although describes a UML extension to support modelling interoperability, it does not define a full methodology for using this notation (Chen et al, 2003). Several methodologies for using UML in the embedded systems domain have been proposed. The most prominent one is UML-RT (UML Real-Time) (Selic and Rumbaugh, 1998), which defines a model with precise execution semantics and can support simulation or synthesis tools, but has limited architecture and performance modelling capabilities.

Another development in the area of embedded systems design and development is that apart from the need to a shift towards system level design, there is also a need for a shift in the item's design as well. As functionality of embedded systems constantly grows and becomes more and more complex, their development cost becomes more and more difficult to be predicted and controlled (Sangiovanni-Vincentelli and Martin, 2001). It has become therefore essential for common architectures for both SW and HW to be developed which allow them to be flexible on implementations and portable between new developments and successive versions of an existing item (Sangiovanni-Vincentelli and Martin, 2001; Sangiovanni-Vincentelli, 2000; Voget, 2003).

These flexible architectures are called ''platforms'' and the concept of designing an electronic item based on platforms is called ''Platform Based Design'' (Sangiovanni-Vincentelli and Martin, 2001; Sangiovanni-Vincentelli, 2000; Voget, 2003). A SW platform is a SW layer that incorporates any standard SW parts, whereas a HW platform is a family of similar architectures that would contain some different components (ie ICs) but would be based on the same microprocessor, so the SW running on these families of similar architectures could be portable. An Application Program Interface (API) is also included to provide for communication between SW and the underlying computational (HW) platform and it is also standardised for each of the HW platforms, so SW can be portable (Sangiovanni-Vincentelli and Martin, 2001; Sangiovanni-Vincentelli, 2000; Voget, 2003). The concept of platform based design can be illustrated using the car's Electronic Control Unit (ECU) as an example. A typical ECU contains (Bouyssounouse and Sifakis, 2005):

- Application and diagnostic SW
- Base SW (RTOS, Network communication)
- Various HW components (microcontrollers, memories, chips, busses, etc).

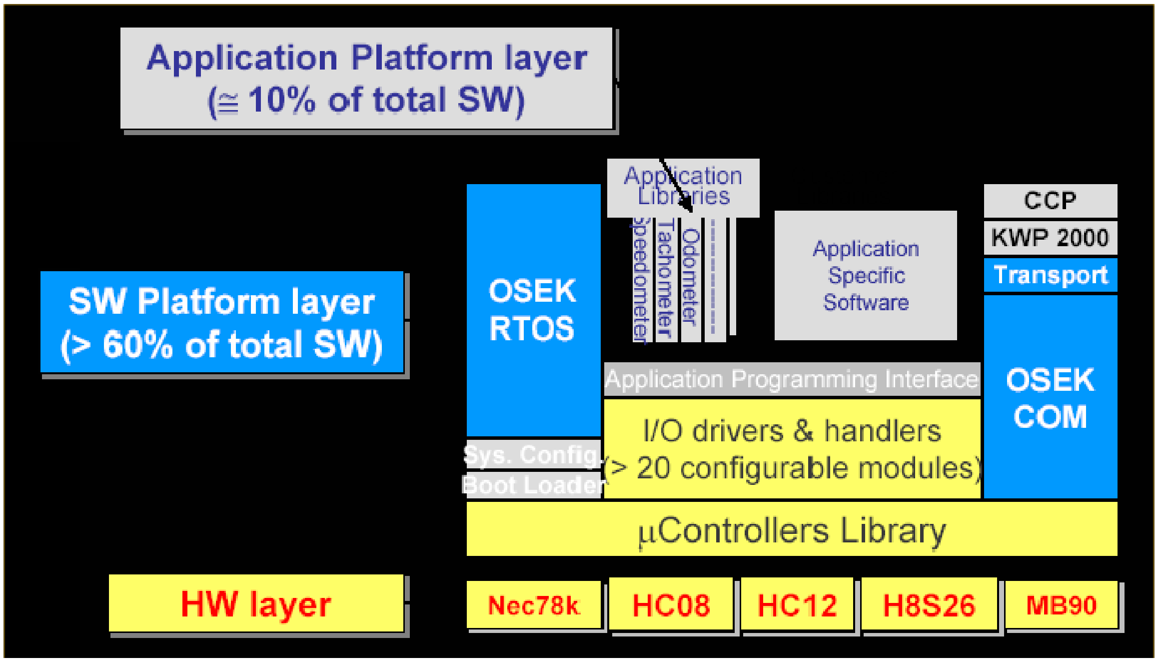


Figure 2.6: A typical car ECU (Sangiovanni-Vincentelli, 2002)

Today, the design and development of an ECU is an ad-hoc process based on the experience and the expertiece of the designer; HW components are being chosen based most of the times in commercial relations whereas SW is very difficult to port from one platform to another (Ziebart, 1991; Bouyssounouse and Sifakis, 2005). Under the platform based design concept, ECU could be transformed as in figure 2.7 with the use of HW and SW platforms:

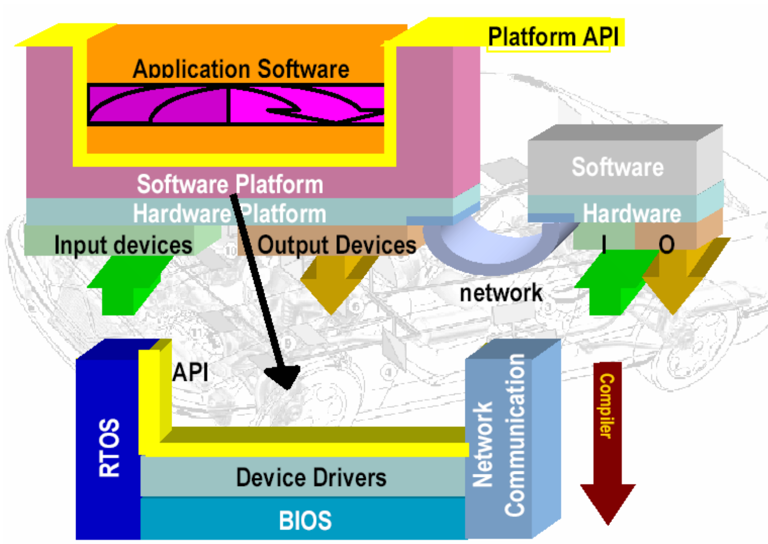


Figure 2.7: A typical ECU (Sangiovanni-Vincentelli, 2000)

Although this approach is yet very new in the embedded systems domain it shows a great potential of substantially reducing design time and cost (Sangiovanni-Vincentelli, 2000; Voget, 2003).

2.1.8: Complexity

As it was concluded earlier (Paragraph 1.1., page 21), "*an embedded system is a complex system consisting of both software and hardware, which is part of a greater architecture and its role is to monitor using its sensors the environment within it operates and to control it, using its actuators".* But what is *complex*? What can be characterised as such?

There have been various definitions developed for complexity; for example, in his book "Programming the Universe" Lloyd (1996) reports 32 complexity definitions. Compexity depends on the domain (ie computer science, neurology, etc) it has to describe. Edmonds (1999) diplays a list with application domains and their coresponding applicable complexity definition and metric. He reviews and evaluates these definitions and concludes that they are either:

- Very broad, that become unusuable when they are applied in practical situtations, or
- Very detailed, which makes them only applicable to specific domain.

In general, complexity definitions aim to describe, at a given time, the properties of a system's ingredients, the properties of the interconnections between the system's ingredients, as well as the properties of –if any- interconnections between the system and its environment. And because systems have different ingredients, properties and interconnections according to what domain they belong to, there could be no uniformal complexity definition.

Another issue, associated with the complexity, which complicates the effort of creating a unique, uniformal definition across domains, is that complexity can not be described uniformily even within the same domain. This happens because complexity depends on the view someone has on the system described. This will be better explain with the following example (Edmonds, 1999):

“Consider three views of a car engine: The first view is of the engine as a single inexplicable unit, a sort of glorified random number generator - it either works or it doesn’t. No explanation is required or deemed relevant; its running is a matter of irreducible luck. If it does not start you ring a mechanic who magically starts it for you. This is the engine as a simple, if malevolent, entity.

The second is a view involving a partial knowledge of the engine. The parts are roughly identified as well as some of the interactions. However these interactions, taken together, are far too complex to understand. If something goes wrong, you can look inside the bonnet and try to identify the cause. Simple experiments in terms of fixing it are possible. Sometimes, with luck, this seems to fix it. Unfortunately, this action often has unforeseen consequences and causes greater long-term damage. When fixing it is beyond this level of understanding, the mechanic is called, who must be (from this viewpoint) a craftsman of deep skill and have a sophisticated understanding of the machine. This is an engine at its most complex.

The third view is that of mechanics. They have an understanding of the decomposition of the engine into functionally near-independent parts. They can use this model to systematically analyse the problem by eliminating possible causes, until the search narrows down to the actual cause. They can then fix this cause, having a good idea of the possible side-effects of this action. This is the engine as manageable complex, due to the appropriateness and utility of the mechanic’s model of it”.

This view was validated in a latter stage (chapter 6), when experts, apart from mentioning that complexity is an important factor in their HW D&D effort estimates, they also noted that there is an issue of complexity within the derived “Complexity factors” (paragraph 6.1) that affect the estimated system’s complexity.

2.2. Embedded Systems in the automotive industry

A car is a complicated mass product, consisting of a number of interconnected subsystems (engine, transmission, etc), which, in their turn consist of hundreds of different components. All these subsystems (and therefore the components) have to work together smoothly and in such a way that they provide the service and the feeling a customer expects from a car (Kopetz, 1995).

Until recently, the electric system of the car was concerned with lights, starter motors, windshield wipers, etc. The situation started to change when car manufacturers started introducing embedded systems to address requirements like safety (ie ABS, Airbag), information to the driver (ie diagnostics), comfort (ie climate control) etc. (Axelsson, 2002). Today electronics are essential to control the car movements, its emissions, to entertain the passengers, etc (Sangiovanni-Vincentelli, 2000). Some cars today contain more than 100 microcontrollers, with many of them working interdependently (Cronenweth, 2004); for example, VOLVO S80 includes ‘’18 ECUs (Electronic Control Units) connected via six networks: a low speed body electronics CAN bus (125kbit/s), a high speed powertrain CAN bus (250kbit/s and four other networks’’ (Cornu and Kung, 2002). In addition, according to Daimler-Chrysler, ‘’electronics will be the source for more than 80% of the innovations in the automotive industry’’ (Sangiovanni-Vincentelli, 2000). Electronics currently account for the 22% of the car and it is predicted that until the end of 2010 this percentage will rise to 35% (EDWARDS, 2003).

In the beginning, the automotive electric system was implemented as it is shown on the left of figure 2.8 below. Each ECU was developed independently from the others and covered one function. For example, one electronic system for fuel injection and another for transmission control (Ziebart, 1991).

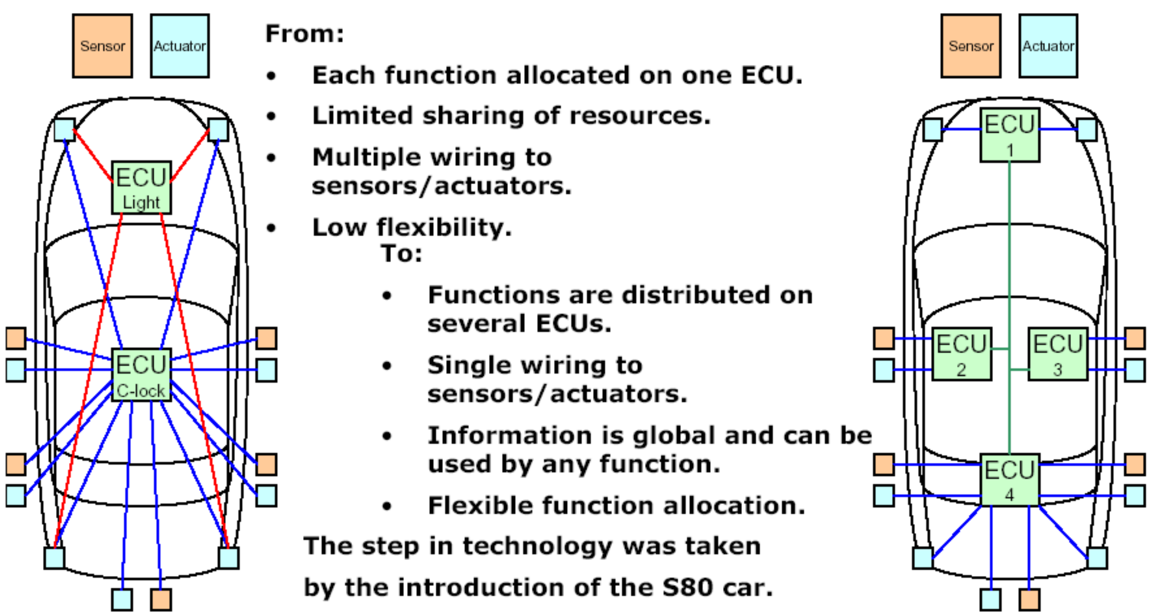


Figure 2.8: The automotive electric system (Butts, 2002)

As the percentage of electronics in the car started to increase, it became a necessity to use functionality distributed in several ECUs. For example, indicator functionality is today distributed over eight ECUs (Voget, 2003a). This made the interconnections between them more complex, and as a consequence, communication networks had to be introduced to reduce the amount and the weight of harness (Ziebart, 1991; Voget, 2003a; Axelsson, 2001). These networks are usually based on the CAN protocol which uses a 250kbit/s transfer rate (Axelsson, 2001).

The electronic items design and development process in the automotive industry is based on the systems-level design and follows the ‘’V’’ design flow, as it presented in figure 2.9 bellow:

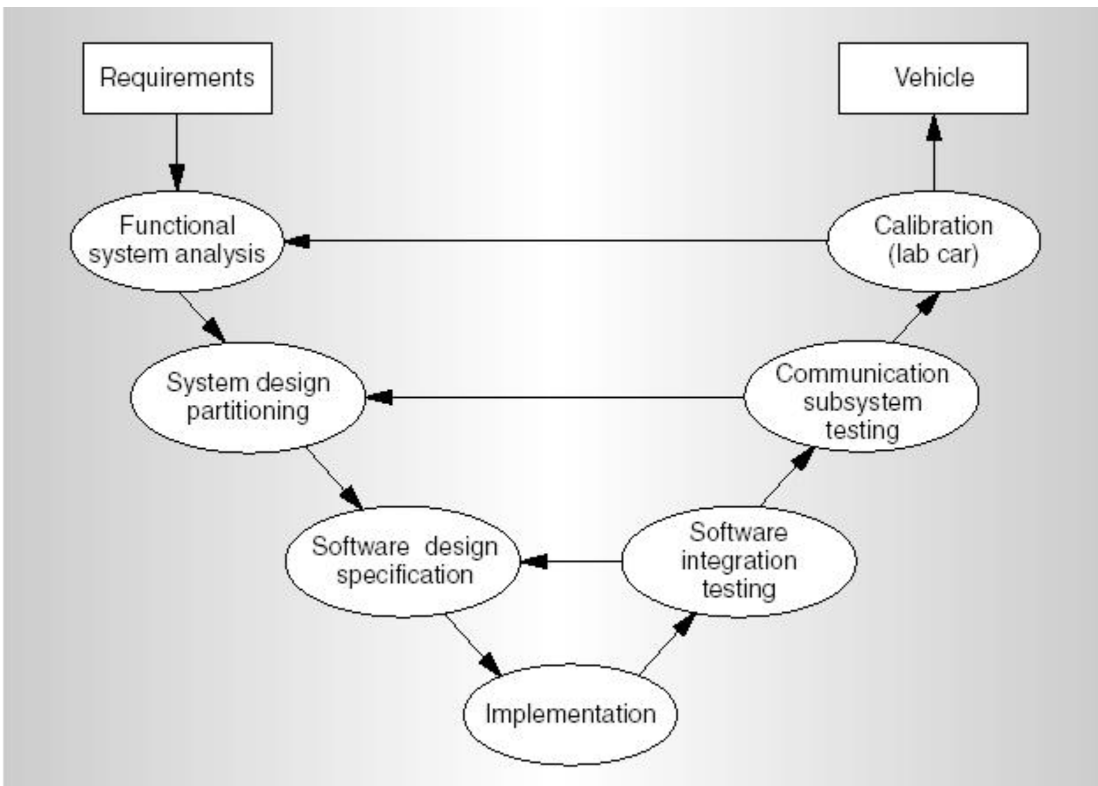


Figure 2.9: ‘’V’’ design flow (Sangiovanni-Vincentelli, 2003)

The development process starts with the system analysis phase, where the designer produces the car’s functional network. In the next phase, the system design partitioning stage, the functional network is distributed over an architectural network and in the SW design specification stage the system designer defines the algorithms of each piece of function. In the implementation phase, the functional network is mapped onto the target HW. The next of the ‘’V’’ cycle describes the validation and verification process (Sangiovanni-Vincentelli, 2003; Bouyssounouse and Sifakis, 2005).

Although a well defined and described process, the above design methodology faces a serious problem, named "Integration" (Sangiovanni-Vincentelli, 2003; Bouyssounouse and Sifakis, 2005; Axelsson, 2001). There are 3 stages during embedded systems D&D that Integration takes place: SW to SW, SW to HW, and when integrating the embedded system in the car's electronic architecture (Sangiovanni-Vincentelli, 2005; Bouyssounouse and Sifakis, 2005; Wagner et al, 2004). Each of these cases are being further presented and analysed in the following sections.

2.2.1. SW to SW

A typical ECU contains (Voget, 2003; Sangiovanni-Vincentelli, 2003; Bouyssounouse and Sifakis, 2005):

- Application and diagnostic SW
- Base SW (RTOS, Network communication)
- Various HW components (microcontrollers, memories, chips, busses, etc).

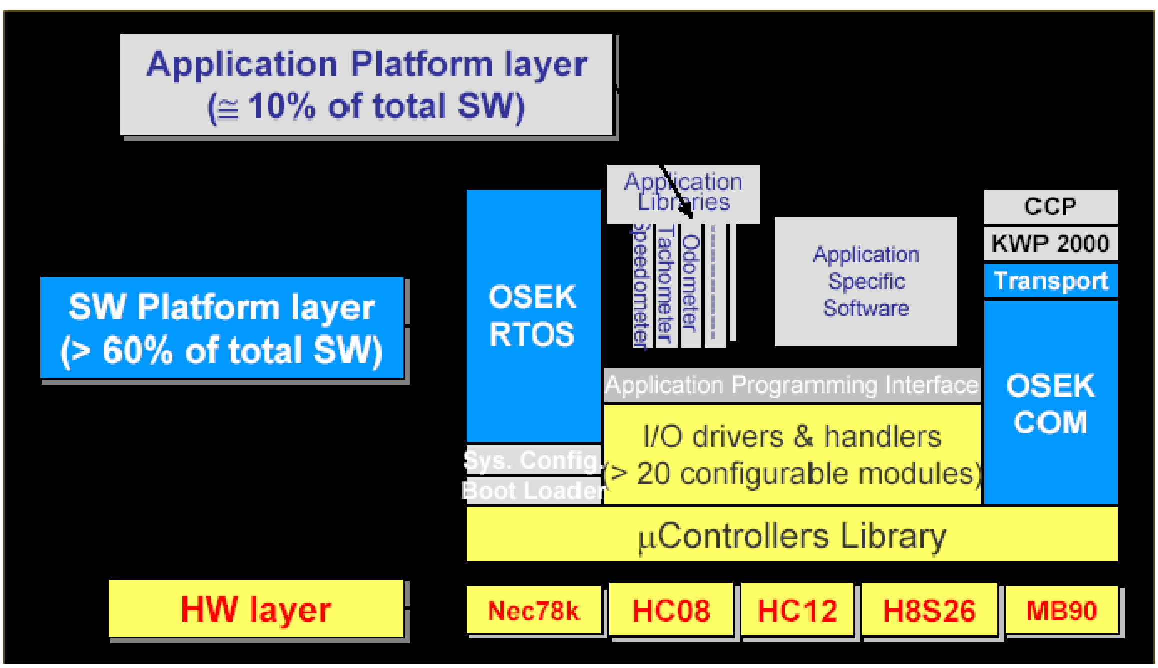


Figure 2.10: A typical car ECU (Sangiovanni-Vincentelli, 2002)

The SW within an ECU consists of 3 layers (Jersak et al, 2003; Voget, 2003):

- 1^{st} layer: RTOS plus I/O
- 2^{nd} layer: Basic SW: Standard functions for the specific automotive application
- 3^{rd} layer: Sophisticated control functions

The ECU supplier outsources the RTOS to an RTOS vendor and then, the ECU supplier adds the basic functions (SW levels 1 and 2). SW layers 1 and 2 are more or less standardised for the specific automotive application (Jersak et al, 2003), whereas the 3^{rd} layer SW is what creates the differentiation in the market between different OEMs (Voget, 2003).

Due to the reason that OEMs do not wish the system's functionality to be exposed, OEMs provide the ECU supplier with detailed design specifications for executable code generation and optimisation for the 3^{rd} layer SW. The integrator is the ECU supplier, since he is the one who has access to all three layers. This leads to numerous interactions between the ECU supplier and the OEM, because the OEM has to verify, each time, the correctness of the ECU and if any problem occurs, it has to send new specifications to the ECU supplier (Jersak et al, 2003).

The application (3^{rd} layer) SW has to be integrated with other SW (RTOS, communication SW, Network SW, etc) contained in the ECU. SW contained in the ECU could be protected by suppliers' IP (like the WindRiver RTOS), which means that their behaviour and structure can not be modified, because this information is not disclosed to the OEM. In this case, additional effort should be consumed for integrating all these SWs by using, for example, APIs or wrappers (Sangiovanni-Vincentelli, 2003; Wagner et al, 2004). Any additional SW modules should be integrated with the application SW in such a way that despite of sharing the same underlying HW platform, it will guarantee the correct system performance (Axelsson, 2001). Three strategies exist for either components integration (either HW to HW or SW to SW): standard-based, IP derivation and communication synthesis (Wagner et al, 2004):

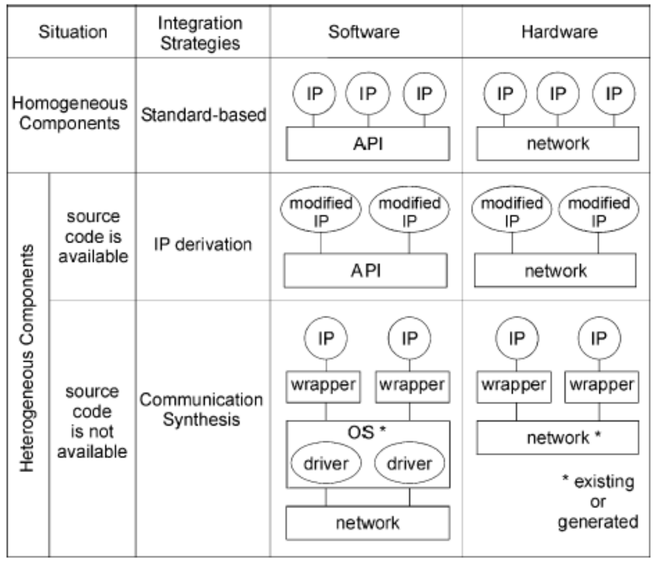


Figure 2.11: Strategies for either components integration (HW to HW or SW to SW) (Wagner et al, 2004)

Table 2.2: Characteristics of the strategies for either HW or SW components integration (Wagner et al, 2004)

Strategy	Description	Problems
Standard-based	The components comply with the same standard, so they directly fit to each other or to a given structure. No adaptation is required.	Many different standards exist.
IP-derivation	The source code of the components is disclosed, therefore the designer can directly access and alter component's functionality.	Alterations depend on each case (specific to each SW or HW component).
Communication Synthesis	The source code of the components is not disclosed, therefore middleware in the form of wrappers has to be created and connected to the components.	The middleware is specific to each individual case (specific to each SW or HW component).

There are two issues in SW to SW integration: timing and memory consumption (Jersak et al, 2003). Timing is concerned with what sequence events are going to be handled by the ES and how much time they will take, whereas memory consumption is concerned with how much memory and for how long each task will occupy (Wirth, 2001; Kopetz, 1997; Berger, 2002; Lee, 2000). Execution time intervals and their sequence have to be determined for each of the RTOS decisions (activate tasks, decide which task will take priority over the others at each given time, how much time this task it will occupy, terminate tasks) and how these decisions will affect the overall sequence of tasks to be executed (Lee, 2000).

Regarding memory allocation, the ECU supplier provides predefined communication variables to SW providers and allocates memory to each one of them, which SW providers have to comply with. If integration problems occur, the ECU supplier has to check each component for its compliance with the settings he has provided and arrange for the appropriate changes to take place. How easy these changes are going to be depends on how the code is written and if it provides for communication and access. If the SW code has been designed in a not flexible architecture, then changes require a lot of effort (Jersak et al, 2003).

Various approaches have been developed to address the above mentioned issues: memory access tables, analysis, simulation, etc (please see paragraph 2.1.3). However, the application of these methods can only be performed by the ECU supplier as he is the system integrator with the OEM having no access to. Integration problems can only be avoided if individual components are designed in such a way that communication variables are clearly defined and either standardised or easy to modify (Jersak et al, 2003; Ziebart, 1991).

2.2.2. SW to HW

Embedded systems consist of a number of HW/SW subsystems, with SW tasks distributed over one or more processors communicating through sophisticated interfaces (Jerraya and Wolf, 2005). Because even today SW and HW continue to be developed concurrently but independently, any communication problems are not detected until the end of the design process (Voros et al, 2003; Axelsson, 1999). As Sangiovanni-Vincentelli states: ''*The issue is the interactions, the side effects that take place. If you are not careful, you can spend twice as long integrating*'' (Edwards, 2001).

The integration of SW with the HW which takes place at the end of the development process is a major electronic items Design and Development cost driver (Sgroi et al, 1999). According to the Data and Analysis Center for SW (DACS) a Department of Defense (DoD) Information Analysis Center, it is very difficult to determine if a failure is due to SW or to HW. The reason for this is (a) the reports generated on the observed failure do not contain enough data to identify the source of the problem, (b) the failure of the system is difficult to be regenerated, and (c) most of the times it is extremely difficult to understand if the failure was caused because of he HW or because the SW was not ‛’intelligent enough’’ to ‛’guide’’ the HW accordingly (DACS, 2003).

The HW/SW co-design approaches presented to date suggest an either SW-centric or a HW-centric based implementation of the system. However, apart from the application SW and the underlying HW, and regardless if the view of the system is HW or SW based, the designers have also to design HW dependent SW and SW dependent HW. Actually, the key problems on embedded systems design are traced back to the HW/SW boundary (Jerraya and Wolf, 2005).

As it was stated before, the key problem on embedded systems design is in the design of the HW/SW boundary, because today’s co-design frameworks are either HW-centric or SW-centric (Jerraya and Wolf, 2005). The strategies for either HW or SW components integration presented in figure 7.2 and table 7.9 solve the problem of integrating HW or SW components, but not the HW/SW interface.

A solution to this problem comes by Jerraya and Wolf (2005) and Wagner et al (2004), who advocate the use co-design based on the ‛’platform-based design’’ paradigm, as this was presented in chapter 2 (paragraph 2.1.6). In their approach, platform-based design starts from the partitioning stage, where interfaces for both HW and SW for the specific target CPU are derived. After that, the procedure continues with HW/SW communication protocol synthesis, HW/SW interconnection synthesis, and at the end, the actual HW/SW interface design as derived from the previous steps based on the specific CPU (shown as ‛’HW adaptation’’ in the (a) step of figure 7.3). Their approach is displayed in the following picture:

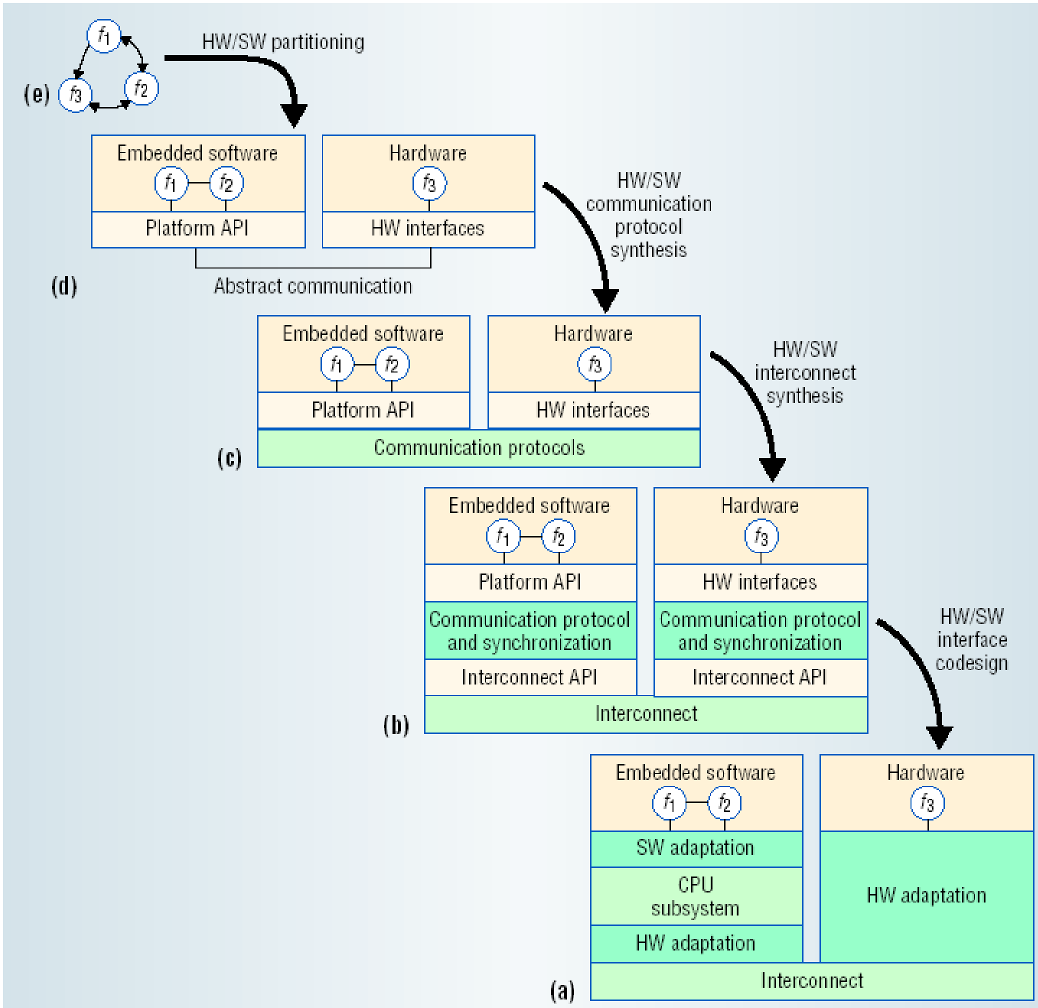


Figure 2.12: HW/SW integration (Jerraya and Wolf, 2005)

2.2.3. Embedded system in the car's electronic architecture

After the embedded system has been designed, it is given to the OEM who integrates it on the car's electronic architecture. Then, the whole car goes through extensive testing and if any problems are detected, then the embedded system is given back to the supplier for modifications. This process is followed until the OEM is satisfied by the embedded system's performance (Sangiovanni-Vincentelli, 2003). This results to major delays in design, in order for these problems to be corrected.

It is observed that when an embedded system is integrated in the car's architecture in most of the cases unpredictable side effects do occur (Bouyssounouse and Sifakis, 2005). This is mainly due to the timing issue: the operation of an embedded system depends on the sequence of tasks it has to perform. However, these tasks come from the embedded system's environment, which, apart from the physical environment, also includes other embedded systems in the car's electronic architecture (Sangiovanni-Vincentelli, 2005).

It has therefore become evident from what has been presented in the previous paragraphs, that producing single systems independently of each other is not the correct approach, since this approach results in major changes when the independently produced by different suppliers systems are integrated in the car's electronic architecture, something that significantly increases both the development and production costs (Axelsson, 1999; Sangiovanni-Vincentelli, 2003; Bouyssounouse and Sifakis, 2005). The fact that the various subsystems are being developed by different suppliers does not allow for an overall distributed system's (car) integrated examination, including the communication links and protocols (Ziebart, 1991; Axelsson, 1999; Sangiovanni-Vincentelli, 2003).

This testing of the complete car's electronic system could only happen if this was performed in a virtual level instead of the actual car at the end of the design process. This would also significantly decrease the development and production costs, it allows for design changes to be introduced every time a test has failed and at for tests to be re-run every time a design change has been performed (Bouyssounouse and Sifakis, 2005). There is also the need for flexible architectures for both SW and HW which allow them to be flexible on implementations and portable between new developments and successive versions of an existing product (Sangiovanni-Vincentelli and Martin, 2001; Sangiovanni-Vincentelli, 2000; Voget, 2003).

In addressing the problems presented above, it is obvious that a cooperation between automotive manufacturers and suppliers has to be established, since the suppliers provide the subsystems, but only the car manufacturer is the one who is aware of the functions and performance of the whole car (Ziebart, 1991; Sangiovanni-Vincentelli, 2000; Voget, 2003). This has resulted in approaches like (Bouyssounouse and Sifakis, 2005; Voget, 2003; Sangiovanni-Vincentelli, 2000; Sangiovanni-Vincentelli, 2003):

- The AEE project: a cooperation between French automotive manufacturers and their suppliers. The project's main output was the AIL (Architecture Implementation Language), a proprietary tool that allows to create or update vehicle databases and eases the exchange of data between suppliers and manufacturers through a common XML format.

- The TITUS project: the same as the AEE project, but for the German automotive manufacturers and suppliers. It was first applied to the body electronics; it has now however expanded in other domains as well.
- The ITEA-EAST/EEA project: a European funded project, with the participation of the major automotive manufacturers and a big number of suppliers, tool-suppliers and research institutes. The aim of the project is to allow for seamless electronic integration through the definition of an open architecture, resulting to a European-wide accepted vehicle embedded architecture.
- A new design methodology is being developed by a number of automotive manufacturers and suppliers, which can be summarised in 3 steps: algorithm specification, virtual prototyping, and physical prototyping.

The major advantages of these methods are that (a) the whole functional network of the car is being examined from the beginning and the specifications of each of the subsystems or items is being extracted as the result of a well defined process including communication and interconnection links, and (b) by integrating and testing the whole structure in a virtual level allows for early detection of problems and immediate test repetition after the problems have been 'virtually' corrected.

2.3. Embedded Systems D&D in other industries

The state of the art on the Design and Development of embedded systems in various industries (automotive, aerospace/defence, consumer electronics, and industrial automation) has been assessed within the framework of the ARTIST project (http://www.artist-embedded.org/Overview). The generic, through industries observations were that

- All industries follow the model based systems engineering design and development approach, and
- The biggest part of the design and development cost is consumed on the integration and testing stage, either due to the fact that individual items and/or subsystems are being developed by different suppliers or due to the lack of appropriate tool support.

Specific differences have been observed within the aerospace/defence industry. In aerospace industry the model based approaches are more advanced and applied than in other industries due to the following reasons (Bouyssounouse and Sifakis, 2005; Preston, 2002; Erdos, 2005):

- The high mission critical applications.
- The fact that systems are extremely complex by nature and it has been proven very expensive to design and maintain.
- The fact that systems should be able to be ported in a number of different platforms.
- The extensive verification and validation and in-flight testing required.
- The need to incorporate in the design the complex human behaviour and its interaction with the complete system.

The design and development of avionics is an active filed with a lot of research going on (ie the SafeAir – http://www.safeair.org/safeair/project/index.htm and Mobies - http://www.rl.af.mil/tech/programs/MoBIES/ projects, and others).

The differences between automotive and aerospace/defence were also observed during an interview between the researcher and the Head of Electronics Design of a major aerospace/defence organisation. This aerospace/defence organisation is one of the two from the aerospace/defence industry that participated in the electronic items cost estimating AS-IS survey (found latter in Chapter 4), it manufactures missile and defence systems and it D&D a limited number of embedded systems in-house. The interview was performed at the organisation's premises, using a semi-structured questionnaire. The aim of this interview was to capture the electronic items D&D process followed in the aerospace/defence industry, the problems it faces and the factors that affect it, as well as if it presents any differences from the automotive industry's D&D process.

The interview was conducted using a semi-structured questionnaire. The questionnaire (which can be found at Appendix B) contains a set of questions that aimed to elicit the knowledge of the expert on how the electronic items D&D process is carried out, but not in a restricted way; it served as a 'guide' of keeping the conversation 'on-track', to allow the expert to expand his views and for the researcher to make follow-up questions based on the received answers, as the expert often raised issues that are not covered by the questionnaire but might be also of significant importance for the researcher in order to have an overall picture of the D&D process. A summary of the expert's answers is presented bellow:

- The D&D process in the aerospace/defence industry follows the ''V-cycle'' approach, as in automotive.
- The effort to D&D an electronic item in the aerospace/defence industry is bigger than in automotive due to (a) stricter and more extensive testing (if an ABS system fails, the driver can still brake, if however the Flight Control fails the consequences could be fatal), and (b) to the selection of components. For example, a fight airplane can experience acceleration up to 10g (10 times the earth's gravitational force). Selected components should be able to undertake this acceleration and accompanied vibration.
- Because of the extensive cost for building and testing a big number of prototypes in order to test an electronic item, testing and validation is being performed using the system's functional mathematical model supported by system test data coming from previous system(s) tests.
- Although a system functional mathematical model is created, there is not a clear process for partitioning SW and HW; this is based on factors like experts' knowledge, company's policies, specific components selection (ie: specific CPU type), etc.
- Integration and testing (a. of HW with SW and b. of the complete item within the system's architecture) is one of the major problems within the D&D of an embedded system. The reason for this is because (a) SW developers do not care of creating the code in such form as to be able to 'guide' the HW but to capture the system's requirements in the code, and (b) HW and SW are being developed independently by people who either understand only the SW part or only the HW part of the embedded system; there are not many people who understand both.
- For an electronic item developed from scratch, 15% of the effort is consumed in HW D&D, 35% in SW D&D and 50% in the Integration and Testing stage, which within the organisation is called 'Proving', since the intention of this phase is to prove that the item works according to its requirements.

2.4. Embedded Systems Specification Approaches

The sponsoring organisation derives the specifications of the ES required, but it outsources its design and development to its suppliers. Specifications are being derived either in the form of UML Use Cases or Statecharts and then passed to suppliers, who are then responsible to provide the required functionality.

2.4.1. Use case based specifications

The Use Case diagram is one of the two main ways for specifying the system's functionality (Vidger, 2003; Maciaszek, 2001; Rumbaugh et al, 2005). It represents the way the system works from the perspective of the system's user (Chen et al, 2003; Axelsson, 2002) and because it is very easy to demonstrate the system's scope, it is easy for the user to understand if the described system offers the indented functionality or not (and therefore validate it or not) (Vidger, 2003; Bell, 2003). The Use Case diagram consists of actors, use cases and associations between them (Vidger, 2003; Maciaszek, 2001; Rumbaugh et al, 2005; Fowler and Scott, 2002, 2001).

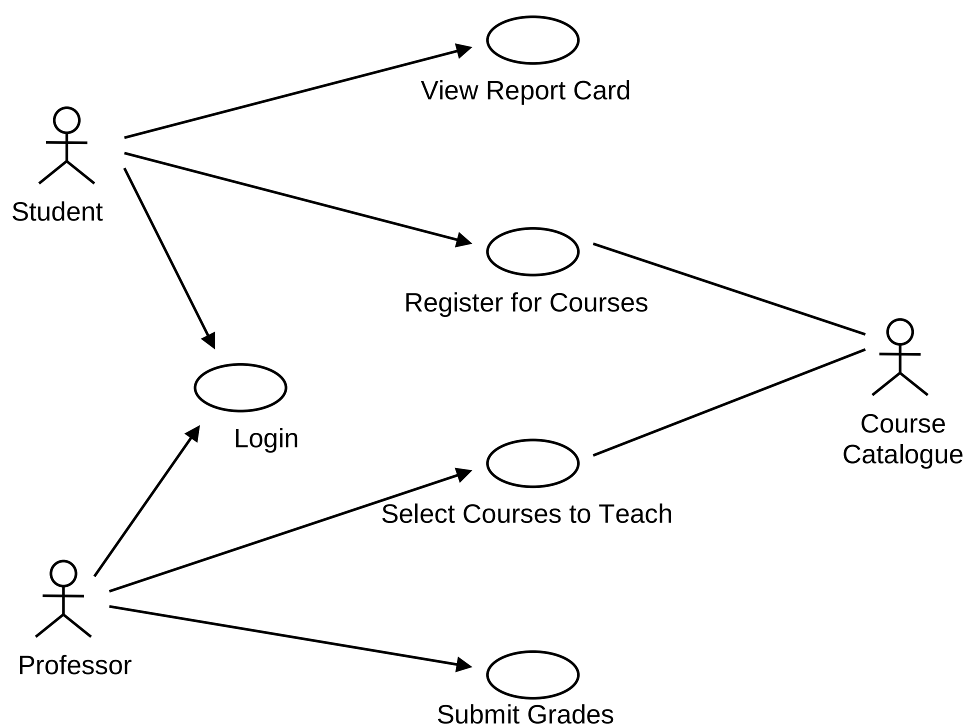


Figure 2.13: A Use Case diagram example

An actor is a role carried out by the user (a physical person or a system) towards the system being described in the use case diagram (Fowler and Scott, 2002, 2001; Rumbaugh et al, 2005). A use case is a unit of the system's functionality (Maciaszek, 2001) and describes in a natural language format the chronological sequence of actions the system performs to produce a specific result on the interest of the actor (Oestereich, 2002). These actions form scenarios which include all the potential alternative outcomes from the interaction between the user and the system (Fowler and Scott, 2002, 2001).

These scenarios are part of the use case's documentation. There is no standard way of documenting use cases; each company/organisation follows its own template (Fowler and Scott, 2002, 2001). However, a general list of what a use case documentation could include is (Maciaszek, 2001; Oestereich, 2002; Fowler and Scott, 2002, 2001):

- A brief description of the use case
- The actors involved
- What should happen for the use case to be initiated
- Detailed scenarios description, including sub-scenarios or alternative scenarios
- A description of the system's situation after use case has been executed.

An example of a use case documentation is shown in the picture bellow:

Use case	Order configured computer
Brief description	This use case allows a Customer to enter a purchase order. This includes providing a shipment and invoice address as well as payment details.
Actors	Customer
Preconditions	Customer points an Internet browser to the computer manufacturer's order entry web page. The page displays the details of a configured computer together with its price.
Main flow	The use case begins when the Customer decides to order the configured computer by choosing the Continue (or similarly named) function when the order details are displayed on the screen. The system requests that the Customer enter the purchase details, including: name of the salesperson (if known), shipment details (customer's name and address), invoice details (if different from shipment details), a payment method (credit card or check), and any comments. The Customer chooses the Purchase (or similarly named) function to send the order to the manufacturer. The system assigns a unique order number and a customer account number to the purchase order and it stores the order information in the database. The system e-mails the order number and the customer number to the Customer, together with all order details, as the confirmation of the order's acceptance.
Alternative flows	The Customer activates the Purchase function before providing all mandatory information. The system displays an error message and it requests that the missing information be supplied. The Customer chooses the Reset (or similarly named) function to revert to an empty purchase form. The system allows the Customer to enter the information again.
Postconditions	If the use case was successful, the purchase order is recorded in the system's database. Otherwise, the system's state is unchanged.

Figure 2.14: Example of use case documentation (Maciaszek, 2001)

A Use Case diagram represents the complete system's functionality. However, in an embedded system SW is the one that provides the functionality and the flexibility required (Ziebart, 1991; Gupta, 2003), whereas hardware (processors, ASICs, etc.) is used as the underlying computational platform (Gupta, 2003). It could therefore be assumed that the Use Case diagram in the embedded system's domain represents the SW functionality.

2.4.2. Statecharts based specifications

Statecharts have been essentially developed to model the concurrent operation of real time systems (Peters and Pedrycz, 1999). The statechart diagram describes the sequence of the different states an object can be found into, as well as how this object advances from state to state (Bell, 2003; Fowler and Scott, 2002, 2001; Maciaszek, 2001; Oestereich, 2002). It is normally attached to a class or an object, but it can also be attached to other modelling concepts (like for example a use case) (Maciaszek, 2001). A statechart diagram consists of (Fowler and Scott, 2002, 2001; Maciaszek, 2001; Oestereich, 2002; Peters and Pedrycz, 1999):

- A set of states, shown in the diagram in figure 2.12 bellow as rounded rectangles,
- A finite set of events, shown in the diagram in figure 2.12 bellow as plain language descriptions (not in square brackets),
- State transitions, shown in the diagram in figure 2.12 bellow as arrows leading from one state to other state(s),
- An initial state, shown in the diagram in figure 2.12 bellow as a black dot, and
- A set of final states, shown in the diagram in figure 2.12 bellow as circled black dots

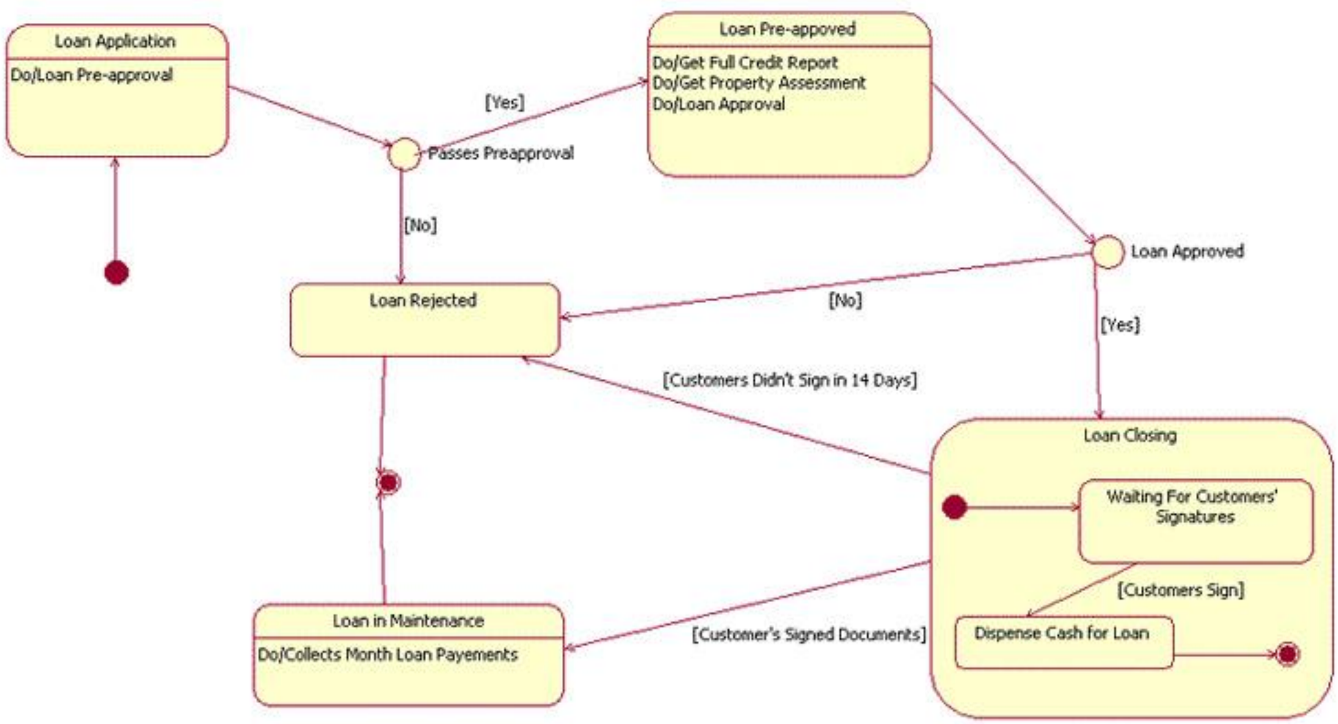


Figure 2.15: A statechart diagram (Bell, 2003)

The object is at its initial state and when the first event occurs, it is triggered to change state and proceed to one of the successive ones. This process is continued until the object reaches the final state.

A Statecharts diagram represents the complete system's functionality. However, in an embedded system SW is the one that provides the functionality and the flexibility required (Ziebart, 1991; Gupta, 2003), whereas hardware (processors, ASICs, etc.) is used as the underlying computational platform (Gupta, 2003). It could therefore be assumed that the Statecharts diagram in the embedded system's domain represents the SW functionality.

2.5. Embedded Systems Cost Estimation

A review in the literature reveals that there is no much work in Embedded Systems cost estimation. In addition, the majority of the literature regarding cost estimating techniques for embedded systems focuses on manufacturing (Graves et al, 1996). Scheffler et al (1998) present a Modular Optimisation Environment (MOE) where a continuous cost-quality trade-off analysis is performed throughout the design and manufacturing process. They note that the main cost factors that contribute to the final cost of an electronic product are:

(i) Indirect costs (machine depreciation, overheads, etc),
(ii) Direct costs (components, process materials, etc), and
(iii) Scrap or rejected cost (either by yield or faults).

They divide the manufacturing process into components definition, carrier (ie PCB) cost, process, assembly, test, and rework or repair. Each potential alternative combination of steps of the manufacturing process is captured in algorithms and at the end, a tool based on Monte Carlo analysis is used to combine all these algorithms in order to derive the final cost of the embedded system. Scheffler et al (1998) approach does not take under consideration the SW D&D cost and covers only the manufacturing process.

Bloch and Ranganathan (1992) present a method of performing cost analysis for manufacturing an embedded system, taking into account the process yield at each step of the process sequence and how the yield at different steps impacts the overall cost of the embedded system. Bloch and Ranganathan (1992) follow a process similar to Scheffler et al (1998): they divide the manufacturing process into sets of states, sets of operations, state transition networks, sets of input processes and sets of outputs. For Bloch and Ranganathan (1992), each action (ie the insertion of a material) has a certain probability to provoke a change to the process state (ie there is a certain probability the inserted material to bond acceptably and another probability not to).

In addition, each process step has a set of its own algorithms that describe the probable outcome for each type of incoming material state. The cost of each process step is determined by a set of inputs of the process model (ie product configuration, number of components per module, material cost, etc) and a set of inputs from process parameters (ie capital equipment, process time, setup time, etc) and plant parameters (ie operator or labour cost, overheads, etc). The effect of each input variable into the embedded system cost is determined using experimental design techniques (ie Taguchi, Plackett-Burman, etc). The experimentl design process results in a set of graphs that is then used to determine the cost of the embedded system based on the set of input parameters. Bloch and Ranganathan (1992) model does not take under consideration the SW D&D cost and covers only the manufacturing process.

Frederic (1997) presents the ‘’Microwave and digital cost analysis model (MADCAM)”. MADCAM is an excel-based model which predicts the production cost of an embedded system as a function of its distinguished design values. MADCAM’s underlying logic is that the cost of any embedded system is the sum of its electronic parts distributed in specific technology categories, times build-up factors that account for the transformation of parts and boards into a finished and tested product. The values of these parameters are defined by the user, who enters these inputs into the model. Then, MADCAM uses an internal computation machine to calculate all the potential alternative manufacturing process steps, based on a database which contains 83 examples of electronic items. Out of these 83 examples, 34 come from space-base systems, 35 come from ship or vehicle systems and the remaining 14 come from airplane systems. As the previous models, MADCAM does not take under consideration the SW D&D cost and covers only the manufacturing process.

As it was sawn earlier in literature review, an ES consists of SW and HW, which are developed independently. Therefore, these 2 different development efforts should be taken into account when the overall ES D&D effort estimation has to be derived. In addition, the Reuse and Integration and Testing efforts should be considered.

Reuse affects the D&D effort of both SW and HW because by reusing parts the developer is able to draw from similar past products and deliver in shorter times than if he was developing the same product from scratch. Additionally, Integration and

Testing is recognized as one of the major drivers in ES D&D. It is estimated that consumes around to 60% of the overall D&D effort (Sgroi et al, 2000). This is because Integration and Testing happens at the end of the development cycle and therefore any problems discovered at that late stage require substantial amount of effort in order to be repaired.

Therefore, ES D&D effort estimation should take into account all these 4 afforementioned entities: HW D&D effort, SW D&D effort, amount of Reuse and amount of Integration and Testing applied. These efforts should be combined in order to create the overall D&D effort. In the following paragraphs, methods/frameworks for estimating HW D&D effort, SW D&D effort, amount of Reuse and amount of Integration and Testing are presented. Their applicability in the embedded systems domain is also examined.

2.5.1. HW

2.5.1.1. Expert Judgement

In Expert judgement, the estimation is based in the estimator's past experience, which means that when an estimator uses Expert Judgement, he has to make a number of assumptions. This, requires that the estimator should have good knowledge of the environment the estimated problem belongs. In other words, it is a highly subjective method (Rush, 2002). Therefore, the quality of the estimation depends on how good the data is, if they are available and how familiar with the estimation domain the estimator is (Hughes, 1996). Sheppard and Schofield (Sheppard and Schofield, 1997) point out that Expert judgement is the most widely used cost estimation method, because (a) it is simple (it does not require any complex algorithms or equations) and (b) it is an integral part of other cost estimation methods (ie: Activity based costing).

2.5.1.2. Detailed Cost Estimation

In detailed cost estimation, the estimator goes through a sequence of steps in order for the complete, overall estimate to be delivered (Stewart et al, 1995). Figure 2.16 bellow describes the process of detailed cost estimating:

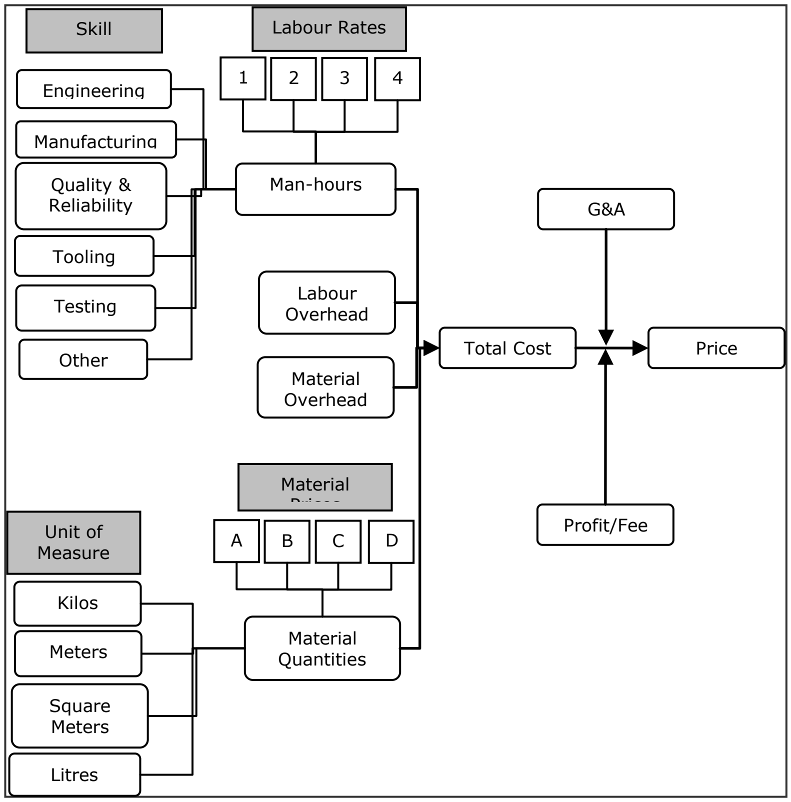


Figure 2.16: Anatomy of detailed estimate (Stewart et al, 1995)

In the detailed cost estimation, for the estimation of labour cost, the hours that each person from each skill is going to be occupied have to be estimated. Then, these hours are multiplied with the corresponding labour rates, in order for their cost to be derived. For the estimation of the material cost, the product of each material's quantity times the material's price per unit is derived. Any labour or material overheads are added to the material and labour cost in order to derive the total cost of the estimated product. Finally, on top of this, administrative costs and profits are added in order for the final item price to be delivered.

Farineau et al, (2001) note that in a detailed cost estimation the estimator, apart from estimating the labour and material costs has to have a good knowledge of the product's domain. This method produces very accurate results, provided that the estimation elements (as they shown in figure…) are known and accurately estimated (Zhang, 1996b).

2.5.1.3. Parametric Cost Estimation

Parametric estimation uses mathematical relationships to estimate costs by creating cost estimating relationships (CER) between the cost of a product and its cost drivers, based on historical data from past projects (Bashir and Thomson, 1999). To describe the principles of parametric cost estimation, Roy (Roy, 2003) uses the following example:

When developing an aircraft, as the mass of the aircraft becomes bigger, so becomes its cost too. If various sets of data of mass versus corresponding cost are depicted in a diagram, then an equation can be drawn between them (see figure 2.17 bellow).

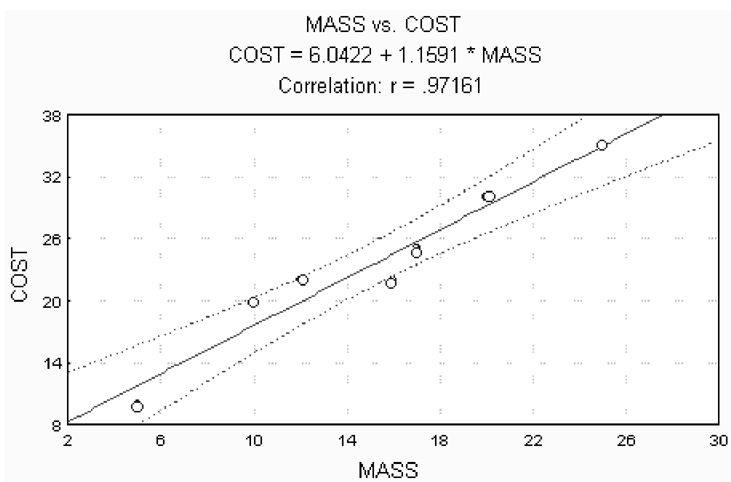


Figure 2.17: Example of a parametric equation (Roy, 2003)

In that figure, the points in the graph represent the relationship between mass and cost for each specific combination of these two. Using correlation, an estimator can derive the mathematical relationship for cost to mass, which, in the example of figure 2.17 is in the form of y = ax + b. Therefore, having this equation derived, the cost for each aircraft of a known mass can be easily derived (Roy, 2003). Variations of this example are the usage of specific ratios to product characteristics. In that case, when ie the characteristic of a product increases, the cost increases to a defined percentage (Farineau, 2001).

As Roy (Roy, 2003) notes, this is a very simple example so as the basic principles of parametric cost estimating to be described; within companies, parametric CERs may become very complex. In this case, algorithms have been developed to resolve these complex situations (Roy, 2003; NASA 1995). CERs are easy and fast to use. Their major disadvantage is that since they are statistical relationships and therefore thay can only show the trend. Also, they require information that sometimes is not available (Duverlie, 1999).

An advancement of parametric cost estimating is feature based costing (FBC) is. The principle of feature based costing is that a feature of the product is used as the basis for the product's cost estimation. The main problem within feature based costing is that there is no standardisation across companies and industries on what constitutes a feature (Rush, 2002). Although standardisation work is currently being done by various institutions (ie ISWG - International Standards Working Group, ICEC - International Cost Engineering Council) (ICEC, 2002), the feature based costing approach is not yet fully established and the implications are not yet completely understood.

2.5.1.4. Activity-Based Costing

Activity-based costing (ABC) relies on the principle that a product is produced by a sequence of well defined activities which have a specific cost per product. The cost of each activity is the product of the cost of the activity per product times the number of how many times this activity is used for the same product. On top of this, the costs that are not directly linked with the product, ie the administrative expenses, are distributed to the activities used to produce the product. Distribution is based on the logic that if an activity is used more than others, it will absorb a bigger part of the indirect expenses than the other activities. Summing up all these costs with direct costs provides the cost estimate (Innes et al, 1994; Zhang, 1996).

ABC provides a more accurate cost estimate than a detailed estimate because it distributes overhead costs more accurately to products which consume resources than the other methods, thus, offering greater accuracy to manufacturing activities such as planning and control (Innes et al, 1994). The disadvantages are that the allocation of costs is based on historical or estimated data and that the allocation of costs to activities is not always accurate (Zhang, 1996).

2.5.1.5. Design-to-Cost

Design to cost tries to find out how the cost of a product can be reduced, without losing its capabilities and keep on performing as it should. It also examines potential trade-offs between cost and functionality within a product, so it can achieve the optimum functionality in the lower cost, while the tight cost targets are met (Vitaliano, 1994). In design-to-cost it is important that the designers give cost the same consideration they give to performance and schedule, considering therefore the cost as a parameter in their designs (Williamson, 1994). In order to do this, engineers need accurate, on-time information, and a way of evaluating the tradeoffs between alternative cost-design implementations (Geiger, 1996).

2.5.1.6. Applicability of the HW cost estimation methods in the ES domain

For most OEMs there are no separate HW specifications. This is because the system's specifications (either in Use Cases or in Statecharts) describe the required functionality of the complete system (''what'' the system should do); however, the realisation of this functionality (''how'' the system would do the requested functions: what resources are going to be used) is the responsibility of the supplier. The supplier decides what resources he will use in order to physically realize the system and achieve the requested functionality; he is the one who decides if he will and how he will implement a function in HW or in SW or as a combination of the two (Bouyssounouse and sifakis, 2005).

Only in a very limited number of cases an additional document, titled ''HW specifications'' is also given to the supplier alongside the system's specifications, which however describes only any non-functional requirements, like number and type of interfaces, voltage, etc. In addition, only for a very limited number of cases OEMs receive back from the supplier a Bill of Materials (BOM) and/or a Cost Breakdown Structure (CBS) which in most of the cases does not include the D&D cost. This leads the engineers and the cost estimators to create proprietary BOM based on their experience and expert knowledge.

Therefore, the information available within OEMS regarding HW contained in an embedded system is:

- Proprietary BOM for a limited number of cases
- A BOM and a CBS for a very limited number of cases, and
- D&D figures, only for a limited number of cases

From what has been presented in the previous paragraphs, it can be concluded that none of the aforementioned HW cost (effort) estimation methods can be applied in the ES domain within the automotive indudtry, because:

- They require information (ie: number of gates, number of hours spent on the project, etc) the automotive industry has no access on.
- This information is with the suppliers and it is not disclosed to the OEM

2.5.2. SW

To minimize overall embedded systems costs, traditional design and test methodologies reduce hardware costs (Debardelaben et al, 1997). Embedded software was considered too small for someone to devote time on it, and has largely been limited by the hardware (Lee, 2000). However, nowadays it is the SW design and development that is one of the major cost sources in the embedded systems design and development (Figure 6.1). Paulin et al (1996; 1997) and Ziebart (1991) estimate that the percentage of an embedded system development effort used for coding is up to 60%, whereas Lavagno et al (1998) estimate it to 70%. This percentage is continuously growing; since SW has shorter development times than HW and because it carries no recurring cost it is perceived easier to alter SW than HW (Douglas, 1999; Humphrey, 1992; Bouyssounouse and Sifakis, 2005). Even many late discovered hardware problems are addressed by altering the software (Humphrey, 1992).

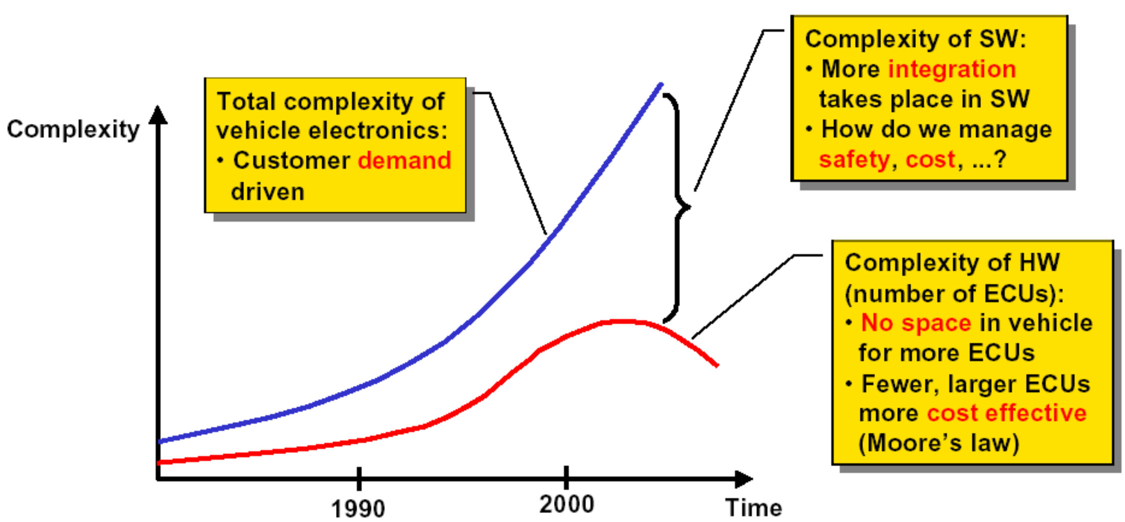


Figure 2.18: Increasing car electronics complexity (Axelsson, 2002b)

It could be argued that embedded SW is SW on small computers (Lee, 2002). However, this view is far from true, because embedded software incorporates some additional features, which, according to Sastry et al (2003), Lee (2002), Karsai et al (2003), Lee (2000), Ball (1996), Sztipanovits and Karsai (2001) and Edwards et al (1997) are:

- Embedded software has the role of configuring the computing device so as to meet the physical requirements instantly (produce the right response on the right time).
- Embedded software coordinates the flow of information through a variety of external sources and the responses given back to them.
- Embedded software should be 'bug free', since under no circumstances it should fail to operate.
- Embedded software must satisfy both logical (computational) and physical requirements simultaneously.
- Embedded systems require extensive testing to assure full-functional operation after the integration of hardware with the software.
- Embedded software often encapsulates domain expertise, particularly when it must process sensor data or control actuators. It is often designed by engineers who are classically trained in the domain.
- Embedded software imposes heterogeneity: control-oriented processes might be mixed under the supervision of a system kernel, which resides on the controller.
- Embedded software has to be integrated with other software (RTOS, EFNOS, etc) and the underlying hardware as well.

All these factors add on the embedded software complexity and make its estimation even more difficult. Karsai et al (2003) mention that the currently existing SW life cycle models focus on the development of the applications software only, and not taking into consideration 'external elements', such as target computer and hardware environment, operating software and other special processes needed for embedded systems.

However, these 'external elements' are developed concurrently with the software, and failing to recognize that fact (understand the complexity of the embedded system) and the interdependence of SW and HW issues, results to programmatic problems (schedule and cost overruns). The development cost is a function of parameters that are very difficult to be quantified, like the ability or the experience of the designers involved, and it is therefore very difficult to provide metrics for this cost (Axelsson, 1997a; Roy et al, 1999a; Roy et al, 1999b; Walkerden and Jeffery, 1996).

Both Vidger and Kark (1994), and Walkerden and Jeffery (1996), classify the SW cost estimating methods either as analogy or model-based.

2.5.2.1. Parametric models

Parametric models generate estimates based on statistical relationships (Cost Estimating Relationships-CERs) by relating dependent variables (i.e., cost and/or schedule) to one or more independent variables (i.e., parameters) (NASA, 1999). A typical parametric cost model is derived using regression analysis on data collected from past software projects. Effort is plotted against the primary cost factor (for example LOC) for a series of projects. The line of best fit is then calculated among the data points (Fenton, 1997).

From all the parametric models, only COCOMO (Constructive Cost Model) offers an embedded software cost estimating module (Boehm, 1981). However, COCOMO has the disadvantage of not covering the requirements analysis and the implementation and maintenance software development life-cycle stages (Boehm, 1981).

On the industry side, the most common parametric models in use for software cost estimation are PRICE-S, offered by Price Systems (www.pricesystems.com) and SEER-SEM, offered by Galorath (www.galorath.com) (NASA, 1999). These tools do not offer specific embedded software cost estimating modules, but they calculate the software cost as part of the embedded system structure, based on few parameters: programming language used, people working on the project, etc and based on their underlying model. These tools have the advantage of using range of values instead of single point values in these estimates.

Parametric models derived by academia (including COCOMO) do not cover the hardware-software integration and testing phase (NASA, 1995; Boehm, 1981), which takes place at the end of an embedded system development. Even if both software and hardware have been tested and found without errors, there is always the possibility for the complete system to present errors in its operation as a whole, because, for example, some SW functions provoke HW problems (or vice versa). Fixing the problem is difficult too, because it may require examining the whole software structure and reworking the code (Ball, 1996). Commercial models (PRICE-S and SEER-SEM) calculate the cost of this stage using parameters, which are included in the estimating models for the embedded system as a whole, based on their underlying parametric model.

Another problem with embedded software is noted by (Vidger and Kark, 1994). They note that in most of the cases of embedded software cost estimating there is little or no system design, which makes it very difficult for the percentage of software to be included within the delivered system to be determined. In addition, the models proposed from academia do not take into account the qualitative issues (Walkerden and Jeffery, 1996; Roy et al, 1999a; 1999b), whereas the commercial models (PRICE-S and SEER-SEM) confront them as parameters, which are included in the estimating models for the embedded system as a whole. Roy et al (1999a; 1999b) conclude that the identification of these issues and their integration provides much more accurate estimates for the design effort.

An additional issue with the commercial models is that they were developed primarily for the aerospace industry, and so, they would not perform well to other environments (Keremer, 1987). In order for these models to be used in other environments, such as automotive, very careful calibration needs to be performed in order to assure that the model produces reasonable estimates (Heemstra, 1992). The same stands for the models derived by academia. For example, as Vidger and Kark (1994) indicate, in the COCOMO model, even by selecting different values for the model's multipliers, results to extreme differences for the maximum and the minimum values of the estimates. Calibration is performed by trying the model using data from past projects, which industry, though, fails to collect (Roy et al, 2000; Vidger and Kark, 1994).

Despite of its usefulness as a parametric tool, the commercial software cost estimating packages are not very popular in industry (Roy et al, 2000; Vidger and Kark, 1994; Heemstra, 1992; Lederer and Prasad, 1993), because:

- Industry uses expert judgment and analogy
- Industry does not collect data from past projects.
- In most of the cases, when industry stated there was a data base with data from past projects available, the database consisted of their memories and the memories of their colleagues.

So, the trend is that commercial models are used to perform sanity checks after an estimate has already been done by using another technique or tool (Vidger and Kark, 1994). Nevertheless, in another domain, the aerospace/defense domain, there is an approach for costing embedded software, suggested by Debardelaben et al (1997), which also takes into account the qualitative and the quantitative issues.

Debardelaben et al (1997) propose an embedded system cost modeling based on a special version of COCOMO model, called Revised COCOMO (REVIC in short) (NASA, 1995). However, this attempt was performed as a customised approach for the production of an Avionic Synthetic Aperture Radar (SAR) for the US Defense Advanced Research Project Agency's RASSP (Rapid Prototyping of Application Specific Digital Signal Processors) program. REVIC adds some additional factors to the original Intermediate Embedded COCOMO model, in order to incorporate the effect of personnel, computer, product and project attributes development and maintenance cost (Debardelaben et al, 1997). The shortcoming with REVIC is that the coefficients it uses have been introduced and calibrated using only DoD past projects (NASA, 1995) and more especially, a variety of military signal processor projects (Debardelaben et al, 1997), and therefore needs calibration before it is applied in another domain.

2.5.2.2. Analogy

Analogy techniques are the simplest form of estimating. Analogy techniques are used to estimate costs by comparing proposed programs with similar, previously completed programs, for which historical information is available. Roy et al (2000), Vidger and Kark (1994), Walkerden and Jeffery, (1996) and Heemstra (1992) identified that industry uses expert judgment and analogy.

Analogy based estimating has 3 advantages over other techniques: it is easy to understand the basis of the estimate, it is extremely useful when the domain is difficult to model, and can be used with partial knowledge of the target project (Jeffery and Walkerden, 1999). Analogy based estimate requires the following steps to be followed (Walkerden and Jeffery, 1996):

- Construction of a representation of the target problem (based on the metrics chosen)
- Retrieval of a suitable case to act as a source analogue
- Transfer of the solution from the source case to target

- Mapping the differences between source and target cases, and
- Adjusting the initial solution to take account of these differences

The analogy software cost estimation requires the creation of database with data from past projects. The history of past projects can be maintained as a computerized database, with detailed metrics and descriptions of characteristics recorded for each project. Using a historical database, an estimator can query the database searching for projects with similar characteristics and then base the estimate on actual costs and process of the previous projects (Cowderoy and Jenkins, 1989). In most of the cases, analogy based estimation is assisted by the knowledge of an expert, and in order to provide meaningful estimates, it needs –as source cases- projects that have been developed in the same environment and belong to the same or a similar product family (Shepperd and Schofield, 1997).

Giannopoulos et al (2003, 2004) review the available SW cost estimating methods and they conclude that none of them is applicable for estimating the embedded software D&D cost in the specifications stage for the following reasons:

1. There is no framework to relate the specifications of an item to its final implementation and its design and development effort.
2. Both analogy and parametric approaches require a database with data (effort, Lines of Code (LOC) or Function Points (FP), etc.) from past projects or access to this. However, industry fails to collect past projects (Roy et al, 2000; Vidger and Kark, 1994) and in addition the details of these projects are kept with the supplier (either protected by IP rights or not being disclosed).
3. From the non-commercial models (COCOMO I and II and REVIC) COCOMO (I and II) offers an embedded mode which however does not cover the integration and testing phase. REVIC does cover it, but REVIC has been developed as a specific approach for the US Defense Avionic Synthetic Aperture Radar (SAR). Both need calibration before they are applied to the automotive domain.
4. Commercial models (SEER and PRICE) have also been developed in domains different than the automotive; they also need to be calibrated.
5. Calibration can not be performed since industry does not retain past projects or projects' information is not disclosed to the OEM by the supplier.

There must therefore be a new framework developed to link the specifications of an electronic item with its design and development effort bypassing this shortage of information, since the specifications are the only information regarding the electronic item the OEM holds.

The majority of automotive OEMs outsource the design and development of their embedded systems. Specifications for the embedded system are being derived either in the form of UML Use Cases or Statecharts (as shown in paragraph 2.4) and then passed to its suppliers, who are then responsible to provide the required functionality. Most OEMs, and only for a very limited number, they receive back from the supplier a Bill of Materials (BOM) and/or a Cost Breakdown Structure (CBS) which in most of the cases does not include the D&D cost (see paragraphs 4.2.3 and 5.3).

Both the Use Cases and the Statecharts diagrams represent the complete system's functionality. However, in an embedded system SW is the one that provides the functionality and the flexibility required (Ziebart, 1991; Gupta, 2003), whereas hardware (processors, ASICs, etc.) is used as the underlying computational platform (Gupta, 2003). It could therefore be assumed that both these diagrams in the embedded system's domain represent the SW functionality.

Various approaches for measuring SW functionality have been suggested: Function Points (FP) (Albrecht, 1979), Mark II Function Points (MarkII) (Symons, 1991), COSMIC Full Function Points (FFP) (COSMIC, 2000) and Use Case Points (UCP) (Karner, 1993). Allan Albrecht (Albrecht, 1979) introduced FP as a way of measuring functionality of application (data processing) SW (Lodeix, 2003; Gao and Lo, 1996) and Symons (Symons, 1991) improved this by creating MarkII FP, which also measures the functionality of application (data processing SW). Latter, Albrecht and Symons joined forces creating COSMIC (The Common Software Measurement International Consortium) and derived the Full Function Points (FFP) method, which counts, apart from application SW, the functionality of embedded SW as well (Londeix, 2003). At the same time, as more and more SW projects are modelled on UML Use Case diagrams, the Use Case Points (UCP) method was developed by Karner (Karner, 1993) to estimate the effort for developing SW whose functionality is captured in a UML model.

All the above aforementioned approaches were considered by the researcher for estimating the effort of D&D embedded SW in the specification stage. A comparison of these methods characteristics, as they come out from the literature review, is displayed in the following matrix:

Table 2.3: Characteristics comparison for SW functionality

	Function Points	**Mark II FP**	**Cosmic FFP**	**UCP**
Supports Application SW or Embedded SW	Application	Application	Both	Both
Easy to use	No	No	No	Yes
Time consuming	Yes	Yes	Yes	No
Requires experienced person for counting	Yes	Yes	Yes	No
Requires information not available in specifications stage	Yes	Yes	Yes	No
Supports UML Use Cases and/or Statecharts	None	None	None	Use Cases

As it is derived by this comparison, none of the Function Point based methods could be applied because of the following reasons (Keremer and Porter, 1992; Jeffery and Stathis, 1996; Lother et al, 2003; Kusumoto et al, 2004; De Tran-Cao et al, 2001):

- FP and MarkII only cover application (data processing) SW.
- All FP-based methods are very time consuming
- All FP-based methods require experienced person for the counting
- All FP-based methods require detailed information about the target SW, which is only available in the detailed SW design specification. This information is not available in the specification stage.

Therefore, the only method remaining is the UCP method. Anda et al (2002), Kusumoto et al (2004) and Ribu (2001) used the UCP method to estimate the effort for SW development, however none of the systems Anda et al (2002), Kusumoto et al (2004) and Ribu (2001) estimated was a real-time embedded system. The researcher decided to go forward and apply the UCP method to estimate the D&D of embedded SW in order to check if UCP could be a reliable estimation method for the embedded SW domain. However, this method would only be applied to specifications expressed in UML Use Case models. Since there is so far no model for estimating development effort from Statecharts specifications, the researcher decided to go forward by developing a new effort estimating framework to cover this domain as well.

2.5.2.3. The Use Case Points method

Gustav Karner (1993) developed a use case based metric, called Use Case Points (UCP) to estimate the effort for designing and developing SW described in Use Case models. The steps on applying the method are described bellow (Ribu, 2001; Anda et al, 2002; Probasco, 2002; Bhushan and Rai, 2003; Kusumoto et al, 2004):

1. Classify the actors of the system and assign weighting factors to them, as shown in the following table (Table 6.2):

Table 2.4: Actor's classification and weighting (Kusumoto et al, 2004)

Type	Description	Factor
Simple	Program interface	1
Average	Interactive, or protocol-driven, interface	2
Complex	Graphical Interface	3

2. Sum each category's actors and multiply them by the corresponding weight. Summing the results derives the Unadjusted Actors' Weight (UAW).

3. Classify the use cases of the system and assign weight factors to them, as shown in the following table (table 6.3):

Table 2.5: Use Cases' classification and weighting (Kusumoto et al, 2004)

Type	Description	Factor
Simple	3 or fewer transactions	5
Average	4 to 7 transactions	10
Complex	More than 7 transactions	15

4. Sum each category's use cases and multiply them by the corresponding weight. Summing the results derives the Unadjusted Use Cases Weight (UUCW).
5. Summing up the UAW and the UUCW derives the Unadjusted Use Case Points (UUCP) value.
6. Calculate the Technical Complexity Factor (TCF) value, which accounts for the overall project complexity. To do this, assign a rating (from '0' to '5' – '0' means the factor has no relevance to the project whereas '5' means the factor is absolutely crucial for the project) to the following list of factors (The weight is defaulted by the UCP method):

Table 2.6: Technical Factors (Kusumoto et al, 2004)

Factor	Description	Weight
T1	Distributed system	2
T2	Response or throughput performance objectives	1
T3	End-user effi ciency (online)	1
T4	Complex internal processing	1
T5	Code must be reusable	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Includes special security features	1
T12	Provides direct access for third parties	1
T13	Special user training facilities are required	1

7. Multiply each weight with the corresponding rating and sum up the results. This gives the Technical Factor value. Then, the Technical Complexity Factor (TCF) is calculated as TCF=0.6+(0.01*TF).

8. Calculate the Environmental Complexity Factor (ECF) value, which accounts for the overall project complexity. The calculation is done exactly as for the Technical Factor (TF):

Table 2.7: Environmental Factors (Kusumoto et al, 2004)

Factor	Description	Weight
F1	Familiar with the Rational Unifi ed Process	1.5
F2	Application experience	0.5
F3	Object-Oriented Experience	1
F4	Lead analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Diffi cult programming language	-1

9. Multiply each weight with the corresponding rating and sum up the results. This gives the Environmental Factor value. Then, the Environmental Complexity Factor (ECF) is calculated as ECF=1.4+(-0.03*EF).

10. Finally, Use Case Points (UCP) are calculated as follows:

    UCP = UUCP*TF*EF.

Various approaches have been suggested in order to convert the UCP to effort. Ribu (2001) notes that initially Kerner (1993) introduced a ratio of 20 hours per UCP and he suggests altering the ratio from 28 to 36 hours according to a combination of the environmental factors. Ribu (2001) also reports that the ratio can vary from 15 to 30 hours per UCP, whereas Probasco (2002) suggests that the ratio is fluctuating from 20 to 30 hours per UCP. These examples show that without past data (productivity figures – effort per use case for each specific organisation), directly converting UCP to effort could be very misleading and it is something that should be avoided (Ribu, 2001).

To overcome this issue, several companies develop custom-made approaches; for example, Sun, IBM, Ericsson and Alcatel have customised the UCP method in such a way to match their organisations' environments (Probasco, 2002; Mohagheri et al, 2005; Anda et al 2002). The common characteristic of these approaches is that all four companies retained data from past projects (effort, number of classes, transactions per class, actual LOC, etc) they developed in-house and therefore were able to correlate these data to UCP elements and modify them, wherever required.

Although very promising, the UCP method has received considerable criticism for the following reasons (Ribu, 2001; Probasco 2002, Anda et al 2002):

- There is no standardised ratio for converting UCP to effort.
- There is no standardised way for writing or structuring Use Cases. That means that different developers, even within the same organisation could produce different Use Cases for the same system.
- Use Cases could also be unbalanced (varying considerably on the number of scenarios they include).
- The use of Technical and Environmental factor introduces additional subjectivity to the method.

2.5.3. Reuse

According to what has been presented so far (Literature and AS-IS models capture), reuse estimation constitutes a major problem when trying to estimate the effort for designing and developing an electronic item because the amount of reuse applied on the design and development of an electronic item (either in the SW or in the HW) can not easily be predicted. The reason for this is that (i) in the specification stage there is no detailed information on how the design is to be implemented and (ii) because this information (implementation details for both SW and HW) is protected by suppliers' IPR. Therefore, none of the above presented metrics (Table…) could be applied within an automotive OEM to estimate the amount of SW and/or HW reuse, since the OEM has no access to the information required for their application. In addition, the metrics are limited to SW reuse only.

2.5.4. Intres

As it was shown earlier, in the literature and it was also validated –latter (chapter 5) - by experts, Integration and Testing effort cannot be predicted, as it is a case-by case issue. This happens because there is no way of knowing before hand if the embedded system's SW might cause a problem in HW – or vice versa – or if the embedded system will work accordingly when integrated in the car's electronic architecture. All the above require additional effort in order to be corrected. However, this depends entirely on the individual embedded system examined.

2.6. Estimating D&D cost

According to Stewart (1995), development cost is the cost of a system up to the point where decision is made to produce an initial increment of the production units or the operational system. Development cost is a function of parameters that are very difficult to be quantified, like the ability or the experience of the designers involved, and it is therefore very difficult to provide metrics for this cost (Axelsson, 1997a; Roy et al, 1999a; Roy et al, 1999b; Walkerden and Jeffery, 1996).

For example, the time an engineer spends on exploring the design space (Design Space: ‘’the process of analysing several ‘correct’ implementation alternatives to determine the most suitable one’’ (Hsieh et al, 2000) in order for the optimum implementation to be found (Vahid and Givargis, 2002; Kalavade end Lee, 1993; Hsieh et al, 2000)), is defined as qualitative design time (Q_L Time) and is not easy to measure because it depends on qualitative issues as the difficulty of the product, engineers’ knowledge and skills, and is most of the times estimated based on the experience and knowledge of engineers with the consequent lack of accuracy (Roy et al, 1999a; Roy et al, 1999b).

Roy et al (1999), Cokins (1998), Heikkinnen (1997), Thompson (1998), Wierda (1990) and Debardelaben et al (1997), note that 70% to 80% of the product cost is committed at the conceptual phase:

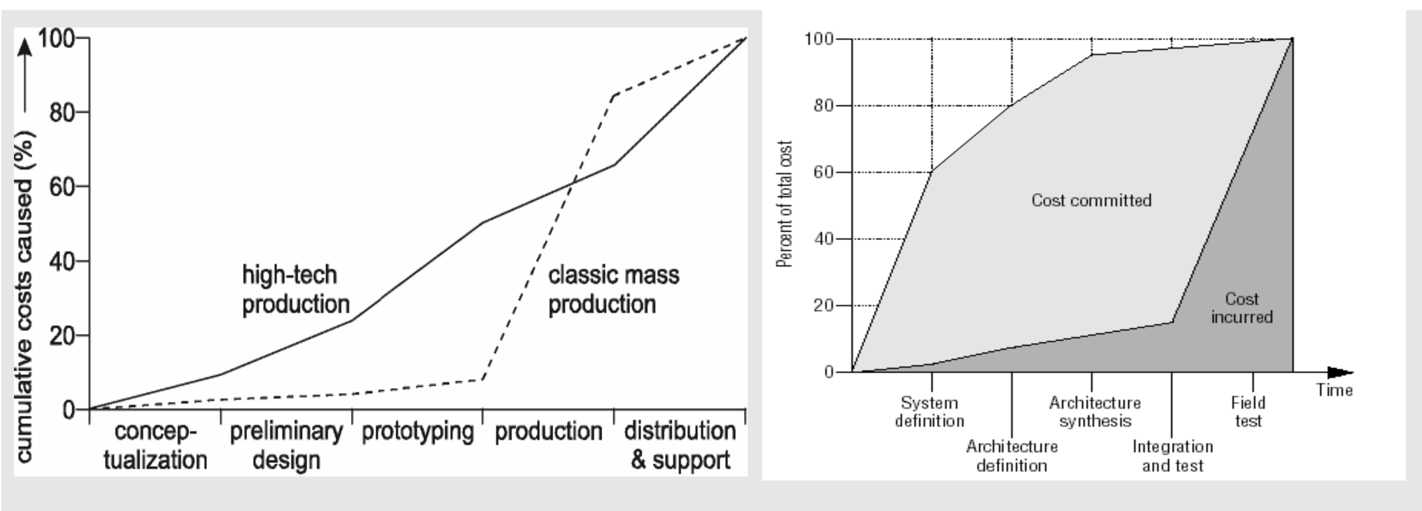


Figure 2.19: Cost commitment over the product lifecycle (Thompson, 1998 and Debardelaben et al, 1997 respectively).

During the design process the product information is not yet available in full detail, so it is difficult to make accurate estimates. However, since design fixes about 70% of the product costs, it is apparent that the most opportunities for cost savings are in the initial system specification and design stage, and therefore, the need for accurate cost estimates in that stage is essential.

Literature on D&D effort estimation is very limited, because of the abstract nature of the task and of the non-easily identifiable items to measure. For that reason, very limited research has been performed in this domain and very few models have been proposed (Bashir and Thomson, 2001). Graves et al (1996) note that design cost can be predicted based on resources utilisation based on a combination of established design standards and from experience gained through the design of similar products. However, these information are company specific, and a Standard for accross any industry could not be predicted.

Roy et al. (1999) developed a methodology for generating Cost Estimation Relationships (CERs) for technical design for a specific aerospace company. Through visits to companies and development of AS-IS models they were able to identify the following qualitative cost drivers (Q^L) and to develop TO-BE models: Material, Manufacturing methodology, Geometric Complexity, Interaction Complexity, Structural Complexity, Overall Complexity, Designer Skills and Knowledge and Redesign level. At the end, the authors developed two different methodologies to estimate non-recurring costs: one which estimates the technical design cost taking into account the data at conceptual phase by relating conceptual definition data with historical data obtained by various part models, and a second one which uses only design activities associated with the development of 3D moles. Both methodologies were tested through some case studies and regardless of performing well, authors concluded that additional testing were required.

A more generic design effort cost estimation model, with the aim to be applicable to a wide range of development environments is suggested by Bashir and Thomson (2001). They define the design effort as the number of hours spent by all the people involved directly in the project and they propose parametric models for estimating it. Although Bashir and Thomson identified a number of factors that create variations design effort (Product complexity, Severity of requirements and use of new technology, Experience, skill and attitude of team members, team size and methods of communication, Use of design assisted tools and Use of formal processes), they present complexity as the major driver and therefore the dominant factor for project size. Complexity is defined as

$$PC = \sum_{j=1}^{l} F_j \cdot j$$

where $PC =$ Product Complexity; $F_j =$ number of functions at level j; $l =$ number of levels, and design effort is estimated as:

- $\hat{E} = aPC^b$ when the project is at the initial stages and few or no data are known.
- $\hat{E} = cPC^d \cdot SR^f$ when the project advances, more data is known and an idea about the severity of the requirements changes can be obtained.

($\hat{E} =$ design effort in hours; $PC =$ product complexity; $SR =$ severity of requirement; $a, b, c, d, f =$ constants).

Bashir and Thomson's model was validated using datasets obtained by two different companies. The model did not performed well on this validation process and Bashir and Thomson concluded that that models for estimating design effort need to be limited to a single company.

Baker (2001), in his approach, develops a hardware cost model for Ford Motor Co. to estimate the total Full Service Suppliers' (FSS) cost for a vehicle/engine program. The author focused his research in two specific areas:

1. FSS R&D cost, since this is the major cost activity incurred by FSS.
2. Costing methodology.

Baker (2001) collected data from Ford's suppliers using questionnaires and personal interviews. When the data were collected, they were grouped and summarised to provide the cost drivers. In the next step, Baker used the same data to improve the first cost model he developed, based on the most significant drivers identified. These were Design Innovation, Engineering Changes, Design Complexity and Manufacturing Complexity. The main output from this research is a cost model, which calculates FSS cost %, FSS value and generates a cost breakdown of the FSS value. Finally, the cost model developed was analysed within workshops composed by the author, Ford's experts and supplier's experts and it was validated with case studies, getting a result with 5% tolerance.

In the DARPA RASSP program, DeBardelaben et al. (1997) developed a design methodology that involved cost-driven architecture selection, task scheduling and performance modeling. For hardware, DeBardelaben et al. (1997) assume that HW cost would be obtained from vendor quotes. For software cost models, they used REVIC (Revised COCOMO) (NASA, 1995). REVIC, a special version of the Intermediate COCOMO model, was developed by US Air Force (NASA, 1995; DeBardelaben et al, 1997) and has the advantage of covering the complete software life cycle, including requirements engineering, implementation and maintenance and hardware and software integration and testing stages (NASA, 1995). This methodology was developed as a customised approach for the production of an Avionic Synthetic Aperture Radar (SAR) for the US Defense Advanced Research Project Agency's RASSP (Rapid Prototyping of Application Specific Digital Signal Processors) program (DeBardelaben et al, 1997), and so, the coefficients it uses have been introduced and calibrated using only Department of Defense's past projects (NASA, 1995) and more especially, a variety of military signal processor projects (DeBardelaben et al, 1997). REVIC model requires calibration before it is applied in another domain.

Axelsson (2002) presents a model for predicting the cost of an embedded system during the design exploration phase, where the design engineer evaluates different system implementations. His model is based for the HW in the individual HW components that make the whole HW infrastructure, where for the SW in the estimation of effort, using Lines of Code (backfiring from Function Points) using the COCOMO model. His model is not a complete solution; it serves however as a good starting point, since it has to be applied to real projects in order to be identified if the proposed cost drivers needs changes or not and for calibration purposes as well (Axelsson, 2002). Axelsson's model does not cover the HW-SW integration, system testing and rework phases.

Ragan et al. (2002) suggest another model for a detailed cost model for concurrent use with HW/SW co-design. Their model is for the initial design space exploration as well. It uses a cost tool, called 'Ghost' which runs alongside the embedded system development. The cost of alternative SW and HW implementations are entered into Ghost, which then calculate the cost of the system for each potential combination of HW and SW implementations. To estimate the size of SW they used a combination of Lines of Code and Feature Points. This model does not cover the SW-HW integration, system testing and rework as well.

Jacome and Lapinskii (1997) proposed a model for estimating the effort for electronics design, which is based on size (gates or transistors) of the objects to be designed, complexity (difficulty of the activity to be performed in a certain environment) and productivity (effort per gate or transistor). For each building block of the design, an effective size is estimated at first. This effective size is then multiplied by a complexity, productivity and reuse factor to produce the design cost for that building block (called Partial NRE cost). The Total NRE cost is a combination of the summation of all the partial NRE costs and of a set of cumulative and non-cumulative activities on the project's hierarchy. The Jacome and Lapinskii model is only applicable for estimating the effort for the HW part of an electronic item.

Design and Development cost is also estimated by commercial packages, like Galorath's SEER-H and PriceSysytems PRICE-H. Both packages use a combination of qualitative (environmental requirements, integration level, etc.), quantitative (size, weight, etc.) and schedule (time to first prototype, amount of new design, etc.) parameters and their underlying databases to estimate both D&D and Life Cycle cost (LCC). In addition, both packages offer special modules (SEER-IC and PRICE-M) for estimated integrated circuit development and manufacturing cost, and special packages for SW effort estimating (SEER-SEM and PRICE-S). These special packages also use a combination of qualitative, quantitative and schedule parameters and their underlying databases to estimate D&D cost for SW and D&D cost and Life Cycle cost (LCC) for ICs.

To estimate the cost of an embedded system, the user has to estimate the cost of SW (using PRICE-S or SEER-SEM) the cost of any ICs (with SEER-IC or PRICE-M) and then import them as add-ins to PRICE-H or SEER-H consequently, in order for the embedded system's cost to be estimated. The problem with these commercial packages is that they were developed based on data coming from the aerospace/defence industry, and therefore careful calibration needs to be performed in order for them to be applied to other domains.

Table 2.8: Design and Development effort estimation models

Framework	What it covers	How it works	To be used in
Roy et al (1999)	Technical design (HW)	Develop CER for technical design by relating conceptual definition data with historical data or by using design activities associated with the development of 3D models	Aerospace
Bashir and Thomson (2001)	D&D effort (HW)	By defining complexity as the sum of the products of the number of functions performed at each D&D level times the number of D&D levels. Then, D&D effort is derived as the product of complexity times an adjustment factor.	Generic
Baker (2001)	Full Service Supplier (FSS) HW cost modelling	From data collected from suppliers, an initial cost model was derived, and then using the same initial data this cost model was refined based on the more significant drivers that were identified during the first step.	Automotive
DeBardelaben et al (1997)	Cost-driven architecture selection, task scheduling, performance modelling	HW cost is obtained by vendor quotes, SW cost is obtained by the Revised COCOMO (REVIC) model. REVIC covers SW/HW Integration and Testing, but has only been applied to very specialised US AirForce projects and therefore it needs calibration before applied in other domains	Aerospace Electronics
Axelsson (2000)	ES Design space exploration	HW cost is obtained by vendor quotes, SW cost is obtained by the COCOMO model. COCOMO does not cover SW/HW Integration and Testing.	Automotive Electronics
Ragan et al (2002)	ES Design space exploration	HW cost is obtained by vendor quotes, SW cost is obtained by the COCOMO model. The cost of alternative HW and SW implementations are fed into a tool called 'Ghost' which then calculates the system's cost. It does not cover the SW/HW Integration and Testing phase.	Electronics
Jacome and Lapinskii (1997)	Electronics design effort (HW)	It is based on the size of the items to be designed, the difficulty of activity to be performed and the productivity. For each building block an effective size is estimated and the total design cost is the sum of the effective sizes times a set of adjustment factors	Electronics
SEER-H and PRICE-H	Electronics D&D effort	Both packages use a combination of qualitative (environmental requirements, integration level, etc.), quantitative (size, weight, etc.) and schedule (time to first prototype, amount of new design, etc.) parameters and their underlying databases to estimate both D&D and Life Cycle cost (LCC). The HW cost is estimated by SEER-H or PRICE-H themselves. The SW cost is obtained by SEER-SEM or PRICE-S, which work the same way as SEER-H or PRICE-H. Both PRICE and SEER also offer specialised modules (SEER-IC or PRICE-M) to estimate the D&D and LCC cost of ICs. Both SEER-H or PRICE-H were developed based on data coming from the aerospace/defence industry, and therefore careful calibration needs to be performed in order for them to be applied to other domains.	Electronics Other

2.7. Gap Analysis

In summary, the researcher identified three gaps: the first is that although Integration and Testing is the major cost driver, the Integration and Testing effort is very difficult to be predicted. There is a lack of understanding about factors that affect Integration and Testing. This is because Integration is a case by case issue and Testing depends on the side effects produced when the item is introduced on the car's complete architecture. Although there are models like the developed by US AirForce REVIC and the commercial SEER and PRICE that cover integration and testing, they are developed and calibrated in domains outside automotive. Therefore, careful calibration has to be performed before they are applied in the automotive domain.

The second gap is about the design and development process itself. Embedded systems design and development is a process that in each stage requires evaluation of various trade-offs. Therefore, a generic effort estimation method would be difficult to be developed, considering the alternatives that could occur.

Finally, the third gap identified by the researcher is that for both SW and HW access to detailed information is required. For HW, the actual cost or design information of the item should be known, whereas for SW access to the code (for obtaining the LOC count) or to the design diagrams (for counting FP) is required. However, access to the required information is most of the times not possible, since this information is property of the supplier and is protected through IP rights and the item's specifications is the only information an OEM holds.

In subsequent chapters, many of these issues are addressed, in particular, the development of a framework that links embedded systems specifications to embedded systems implementations and cost, in order to by-pass the suppliers' ''IP blockage''. The following chapter presents the research strategy deployed by the researcher in order to successfully achieve the aim of this research.

2.8. Summary and Key Observations

In Section 2.1, the researcher presents the design and development process for an embedded system. The presentation starts with the traditional design, goes through co-design, model based system co-design, co-design with IPs and reusable components and concludes with the latest developments in the ES D&D domain. The key observations within this section are the following:

• Traditional design approach seriously limits the ‘design space’.
• In traditional design, HW and SW are developed separately. This, results in SW/HW
• Integration was performed only at the end of the development process, requiring costly revisions and excessive rework to be applied.
• Co-designing SW and HW results in fewer integration problems at the end of the design process. However, co-design enters the development process in a stage where important implementation decisions have already been made. This incorporates the danger of co-designing a solution which is not the optimum one.
• With system-level, model based design, a model of the complete system’s functionality is being produced, and the final implementation is the result of a well established and structured process. This process is validated against all design constraints in each of the design stages, ensuring that the final implementation is the optimum implementation (‘’correct by construction’’).
• The importance of IP protected, reusable blocks, either in SW or HW has been acknowledged and methodologies based on their use have been presented.
• Although various co-design frameworks exist, they are rarely used by companies, who usually develop their internal methodologies. The reason for this is that there are various parameters on the design process, like the experience or the expertiece of the designer, that dictate the design solutions.
• The use of UML as a System Level Description language. Through its diagrams, UML can capture the complete system’s behaviour in various stages/phases, making it a de-facto standard for system level design.
• The advent of ‘platform-based design’. SW and HW are being customised in such a way that it is very easy to be ported through similar implementations or successive versions of the same product.

In Sections 2.2 and 2.3 the researcher presents the current status of embedded systems usage in the automotive and other industries as well. The key observations are:

a. in automotive:

•	The development process follows the V-cycle, system level approach
•	There is a move from each function allocated on one ECU to various functionalities distributed over various ECUs.
•	Until recently, various different suppliers were developing independent systems. This leads to serious integration problems, when these independent items are incorporated in the car’s architecture.
•	To overcome the above issue, there are recent attempts for the development of visual platforms, where the complete behaviour of the car’s electronic structure is being described and tested in a virtual level, before the actual implementation is performed.

b. in other industries:

▪	Other industries are also following the model based systems engineering approach.
▪	The biggest problem is the integration problem
▪	Differences are being observed in the aerospace/defence industry, where due to the nature of the application extensive verification, validation and in-flight testing is required.
▪	Due to the nature of the application, model based approaches are more advanced within the aerospace/defence industry.

In Section 2.5, the researcher argues about the significance of the design and development cost. Since 70% to 80% of the product cost is committed in the design and development stage, the need for accurately predicting the design and development effort (and therefore the cost) of an embedded system is crucial. Existing approaches to embedded systems design and development effort estimation are being presented, and the following can be observed:

▪ Most of the presented frameworks cover only the HW part of an ES
▪ Frameworks that cover both HW and SW require access to information (like nuber of gates for HW or Lines of Code –LOC- for SW) that is not available on the specification stage.
▪ Most of the frameworks do not cover the Integration and Testing effort, which is the major cost driver in the D&D of embedded systems. The one framework that does cover it, require access to Lines of Code (LOC).
▪The HW cost of the embedded system is obtained either by the actual components costs or through design characteristics (ie gate counts).
▪The SW cost of the embedded system is obtained by using the COCOMO model or one of its variations, the REVIC model. If access to the code is granted, then Lines of Code (LOCs) are being used as an input metric to the SW models, otherwise, Function Points (FP) are calculated from the SW design diagrams and then converted to LOC.
▪Access to information necessary for estimating D&D effort (ie LOC or number of gates) is not available to the OEMs, since they outsource the D&D of their ES, and this information is protected by supplier’s IP rights.
▪The importance of the qualitative design time (Q^L – the time the engineer consumes on evaluating alternative decisions exploring the ‘design space’) has been acknowledged. However, this time is very difficult to be estimated, because it depends on qualitative issues like engineer’s knowledge and skills.

In summary, the author has presented a structured account of the literature regarding ES D&D within automotive and other industries, as well as of various approaches for estimating the D&D effort. The author identified a lack of research as to the type of data and information that is necessary in order to perform cost estimates for the automotive industry. It is the author’s intention to contact an AS IS study (Chapter 4) to address these issues. The author also identified the absence of a framework to link the ES specifications with its D&D effort. It is the author’s intention to investigate that further and provide this kind of framework for the automotive industry.

To successfully achieve this aim, an appropriate research methodology is designed. The following Chapter focuses on the development of a research strategy to address many of the key issues highlighted within this structured account of literature.

3. Research Aim, Objectives and Methodology

In the previous chapter, the ES D&D processes throughout various industries and a review of approaches regarding ES D&D cost estimating were presented. The key observations from this study were:

- There is very limited research in the area of ES D&D cost estimating.
- There is no framework/approach linking the ES specifications with its D&D cost.
- Most of the frameworks/approaches presented in the literature cover only the HW part of the ES D&D.
- All the presented frameworks require access to information (ie LOC, number of gates, etc) which (a) is not available at the design stage and (b) it is protected by supplier's IP rights, and therefore it is not accessible by the OEMs.

These key observations served as the basis for the researcher to set the aim of the research and its associated objectives. In this chapter the researcher presents the choices he performed in order to derive an appropriate methodology which will help him to accomplish the aim and the objectives set.

To achieve this aim, the chapter is divided into several sections. In section 3.1, the research aim and objectives are defined. In section 3.2, available research approaches and research strategies are considered. Due to the exploratory nature of this research, a qualitative research approach is chosen. A survey study was deemed most appropriate for the initial research and a case study research approach was used to develop further the ideas defined. In section 3.3, the research methodology is defined and the research plan is illustrated, which shows how risks to research validity are countered. In section 3.4, chapter summary and key observations are provided before moving onto chapter 4, which discusses how the initial results were collected.

3.1. Research Aim and Objectives

The aim of this thesis, stated earlier within the introduction, is:

''to formalise the cost estimating process for the Design and Development of embedded systems in the specifications stage in the automotive industry''

The relevance of this research is more clearly visible now that the literature review in chapter 2 has been completed and the research 'gaps' identified:

- Estimating the effort (and therefore the cost) of D&D an electronic item has become of absolute importance for the automotive companies.
- There is no access to the necessary information on D&D effort, since this is property of the supplier and it is protected through IP rights.
- The need for the development of a framework/model/approach/method to connect the high level requirements or the specifications of an electronic item with the cost estimating process in order to 'by-pass' the supplier's 'blockage' has been identified.

To address these research gaps, the following research questions are raised:

- What type of data do electronic parts cost estimators need to assess the D&D cost of an electronic item during the specification stage?
- What level of detail is required?
- How can these data be modelled and linked with the product cost estimates?
- Is it possible to create a model/framework/method/approach to facilitate this process?

By answering these questions, the author seeks to address the lack of research related to estimating the D&D cost of an electronic item, when the only information available is the item's high level requirements or specifications. In addition, to assess whether a model can be developed to model this process and therefore create a structured and formalised cost estimating process. This is a multidisciplinary research, considering HW, SW, Reuse and Integration cost. To guide the research process the following objectives were set:

- To identify state of the art research in electronic parts cost estimation and in particular on estimating their D&D cost within the manufacturing industry.
- To assess the challenges organisations face when estimating the D&D cost of an electronic item within the automotive industry.
- To link the specifications of an electronic item with the cost estimating process.
- To develop a framework to structure and formalise the cost estimating process within an automotive OEM for ES D&D. The framework will include HW, SW, Reuse issues and their Integration and Testing.

3.2. Overview of Research Methodology

Once the research aim and objectives are defined, the next step on designing the methodology to be applied is to identify the research purpose and the research strategy to be used.

3.2.1. Research Purpose

Robson (2002) defines three different kinds of purposes for carrying out research: exploratory, descriptive or explanatory (Table 3.1).

Table 3.1: The purposes of research (Robson, 2002)

Exploratory	This has the aim of analysing a new or unknown subject, asking questions in order to collect information about it. As the subject is new, it is not possible to collect quantitative data. Therefore, usually the data collected are qualitative data.
Descriptive	This has the purpose to provide a profile of an established situation, requiring a substantial knowledge of the situation. The required knowledge allows the researcher to select the kind of data that he needs to collect. The data collected can be both qualitative and quantitative.
Explanatory	This gives an explanation to an existing problem in more situations, usually in the form of causal relationships. The data collected can be both quantitative and qualitative.

The previous chapter concluded by identifying the need for research into the electronic items D&D process, especially on the issue of estimating the effort devoted to design and develop an item based only on its high-level requirements or its specifications. There is a small quantity of public information about this concept and all the issues related. For this reason, the purpose that best matches with this research is the *Exploratory* one.

Exploratory research almost exclusively requires a flexible or qualitative approach. It is a form of enquiry that seeks to find new insights, asks new questions, assesses phenomena in a new light, or generates ideas and hypotheses for future research (Robson, 2002).

3.2.2. Selecting a Quantitative or Qualitative Research Approach

Quantitative research is often referred to as the traditional scientific research approach. It is considered as a pervasive, scientific mode of enquiry, characterised by objectivity, reliability, and prediction. Much of the data collected and used is of a numerical format. Quantitative researchers tend to be dispassionate, neutral, and detached when using this form of enquiry. The most common form of this research approach is within laboratory settings, where the environment and experimental conditions can be closely controlled.

The main strengths of the quantitative approach lie in precision and control. Control is achieved through the sampling and design; precision is achieved through quantitative and reliable measurement. The main limitation, with respect to 'real world enquiries', is that human beings are far more complex than the 'narrow' view imposed by a quantitative approach. Unlike within a laboratory setting, it is difficult to gain control over the many variables within a social setting (Burns, 2000).

Qualitative research is primarily based on an investigative approach, where much of the data collected is through interviews, surveys, and observation, and is in the form of words (Robson, 2002). Qualitative researchers tend to be personally involved with their study. As a result, the research questions and design tends to 'evolve' over time as more information is collected. Sociologists, psychologists, anthropologists, and more recently business and industry, tend to use a qualitative research approach (Gummesson, 1991).

The main strengths of the qualitative research approach are the insights gained from an inside view of the world under investigation and the researcher's personal involvement. This enables the researcher to derive unexpected and striking observations to examine further. The main limitations and criticisms are validity and reliability. Data collection methods are time consuming, subjective and prone to interpretation bias. The fact that the researcher is present causes change and bias during the collection of data. It is difficult to replicate studies; furthermore, it is difficult to make generalisations from the research findings. Nonetheless, since the 1960's and 1970's, qualitative research strategies have been gaining more credence, providing powerful tools for research in management and business administration (Gummesson, 1991).

Since the purpose of the research has been defined as Exploratory, the qualitative approach is to be used, since qualitative approach and Exploratory research are almost exclusively linked (Robson, 2002).

3.2.3. Research Strategy

There are many research strategies or methods that can be used to collect the data necessary to answer the research question. Robson (2000) presents three traditional research methods widely used and recognised: Experiments, Surveys and Case Studies:

Table 3.2: Characteristics of research strategies (Robson, 2002)

Experiments	"Measuring the effect of manipulating one variable on another variable. Its typical features are: (i) The selection of samples of individuals from known populations, (ii) Allocation of samples to different experimental conditions, (iii) Introduction of planned change on one or more variables, (iv) Measurement on small number of variables, (v) Control of other variables, and (vi) Usually involves hypothesis testing."
Surveys	"Collection of information in standardised form from groups of people. Its typical features are: (i) Selection of samples of individuals from known populations, (ii) Collection of relatively small amount of data in standardised form from each individual, and (iii) Usually employs questionnaire or structured interview."
Case studies	"Development of detailed, intensive knowledge about a single 'case', or of a small number of related 'cases'. Its typical features are: (i) Selection of a single case or a small number of related cases of a situation, individual or group of interest or concern, (ii) Study of the case in the context, (iii) Collection of information via a range of data collection, and (vi) Techniques including observation, interview and documentary analysis."

The research is performed using two different research strategies: surveys and case studies. Survey was used in developing the AS-IS models (current practice) on electronic items cost estimating process in a number of companies and industries. The AS-IS study is presented in detail in the following chapter (chapter 4). Surveys were also used for the developed of the cost estimating model (presented in chapters 5,6 and 7). Multiple case studies were used in the main body of the research, in the model's development stage. Case studies were chosen for two reasons:

1. The research is sponsored by industry, so case studies are available and they provide access to 'real world' situations. Case studies, combined with the literature review and the AS-IS models' results, provide the basis for exploring ideas and developing models.
2. Secondly, case studies, within a business related context, tend to be limited to exploratory research (Gummesson, 1991).

Next section presents the course of action the researcher adopted on collecting case studies and performing their analysis.

3.3. Case-study Strategy

Case-study research is increasingly accepted as a scientific tool for industrial research. In many countries, doctoral theses are using this approach for topics such as marketing, strategy, and analysis of organisations (Gummesson, 1991). One of the main purposes of case studies is their use as a preliminary study to major investigations. They are often used as a source of hypotheses for future research by showing that things are so, or that such an interpretation is plausible in a particular case and therefore might be so in other cases. In addition, they are often used to refute universal generalisations. As a result, conclusions are instrumental rather than terminal (Burns, 2000).

3.3.1. Case Studies and Data Collection

The case study principles of data collection are to use multiple sources, maintain a chain of evidence, and record the data (Burns, 2000). The main techniques are observation (both participant and participant observation), interviewing (informal, structured, and unstructured), and document analysis (Gummesson, 1991; Burns, 2000). Yin (1994) states that no single method has a complete advantage over all the others. In fact, the various sources are highly complimentary, and that a good case study will use as many sources as possible.

This research adopts the case study principles by designing in the use of multiple sources of data collection.

3.3.2. Case Study Issues

Although gaining acceptance within the research community, the case study approach is still viewed by some as a less than desirable form of enquiry (Burns, 2000; Gummesson, 1991; Robson, 2002). The main issue is related to trustworthiness, which translates into several concerns such as: human subjectivity, generalisation, time and information overload, reliability, validity, and rigour. These issues affect all qualitative research strategies to greater or lesser extents; nonetheless, there are strategies one can use to counter such risks.

3.3.2.1. Validity and Reactivity

Validity refers to the issue of how well a method/framework/model represents the reality (Gummesson, 1991). The main threats to validity in flexible research are reactivity, respondent bias and researcher bias (Robson, 2002). Reactivity refers to how the researcher affects the case study deployment by his presence, respondent bias refers to how the respondent’s background or personal opinion may affect the case study outcome (ie the respondent presents the researcher what the researcher would like to hear and not his true opinion), and researcher bias refers to how the researcher’s background or personal beliefs may affect the case study outcome. Robson (2002) categorises the ways to confront these threats as:

Table 3.3: Strategies for Dealing with Risks to Research Validity (Robson, 2002)

	Risks to validity		
Strategy	**Reactivity**	**Researcher Bias**	**Respondent Bias**
Prolonged involvement	Reduces threat	Increases Threat	Reduces threat
Triangulation	Reduces threat	Reduces threat	Reduces threat
Peer debriefing and support	No effect	Reduces threat	No effect
Member checking	Reduces threat	Reduces threat	Reduces threat
Negative case analysis	No effect	Reduces threat	No effect
Audit trail	No effect	Reduces threat	No effect

The researcher used a number of the above mentioned techniques in order to ensure the validity of his research. However, the data collection process was difficult, due to the limited access to people. A full record of activities carried out during performing the research was kept in the form of documents, emails, audio tapes, research journal (audit trail). The materials collected (whatever their form – documents, transcripts, interpretations by the researcher, etc) were presented back to the respondents and they were asked to check them for their correctness (member checking).

Meetings with the participants as well as other experts were held within the research's environment, in order for the researcher to understand that he holds a comprehensive picture of the current situation, the inputs received and the interpretations given (peer debriefing and support). The AS-IS capture interviews were taken jointly with another researcher (Mr. Aitor Sarasua-Echeverria), however, the analysis of data was performed separately, since Mr. Sarasua-Echeverria concentrated on analysing and presenting the AS-IS on Printed Circuit Boards (PCB) D&D cost estimation. Finally, multiple sources of data collection were used (documents, interviews, workshops, observation, models' analysis, etc) in order for the researcher to enhance the fidelity of the research (Triangulation). Prolonged involvement and negative case analysis were not used in this research.

3.3.2.2. Reliability and Replicability

Reliability refers on the issue of if two or more different people performing the same research will derive the same outcome (therefore very related to replicability) (Gummesson, 1991), which something rarely achievable in the qualitative research (Robson 2002). Ways to overcome this obstacle are triangulation, keeping an audit trail and being aware of potential areas of researcher bias (Burns 2000; Robson, 2002).

3.3.2.3. Generalisability

Generalisability is concerned with if the research findings are applicable to other populations as well, apart from the population they originated. In most of the cases, a case study is chosen because it is available or because it is unique (Burns, 2000). In addition, it is also common that a small number of cases is used in order for the findings to be developed (Gummesson, 1991).

3.4. Research Methodology

Based on the research choices made, and developing an understanding of the issues related to undertaking a qualitative case-study research strategy, a research methodology is proposed in Figure 3.1. The research methodology is divided into three phases namely: 1) research strategy development 2) data collection and idea formation 3) data analysis and validation.

In the first phase, *'research strategy development'*, the purpose was to decide on an appropriate research strategy. Quantitative and qualitative research approaches were considered and a qualitative strategy chosen. After which time, available research strategies were considered (see table 3.2). The methods and issues related to the chosen research strategy i.e. case study were analysed to understand how best to use this strategy.

In the second phase, *'data collection and idea formation'* the data collection techniques and risks to validity are considered and planned, as they are presented in paragraph 3.3 above. The qualitative research approach chosen by the researcher carries a number of potential dangers regarding the collection and interpretation of data as well as regarding the research's validity. To overcome these dangers, the researcher employed a combination of counter strategies, as they are described in paragraph 3.3.2 above.

For example, the author time within the sponsoring organisation during the AS IS study environment. Using a survey approach based on a semi-structured questionnaire and strategies for encountering any threads to validity (described at table 3.3) the researcher was able to create the AS-IS models presented in chapter 4. This 'prolonged involvement' helped overcome many of the issues related to case study research mentioned earlier in section 3.3. Spending such a long time within the environment has the disadvantage of introducing researcher bias. To counter such a threat, multiple sources of data collection techniques were used. For example, participant observation, interviews, process models and document analysis. Using a combination of data collection techniques helped provide validity and reliability to the results. The specific techniques for performing the AS-IS capture and the AS-IS results are fully discussed and presented in Chapter 4.

In addition to collecting data from within the case study environment, available literature was constantly reviewed. This helped to develop ideas and relate theoretical issues to the 'real world' case study environment and vice-versa. Using a combination of literature, case studies and semi-structured survey and employing again the strategies of table 3.3, the researcher was able to create a model for estimating the effort for Designing and Developing an Embedded System based on its specifications (see Chapters 5, 6 and 7).

The closing stage of the research methodology was the *'model validation'* phase (Chapters 5, 6, 7 and 8). Initially, the developed framework was validated with electronic cost estimating experts from within the sponsoring organisation (Chapters 5, 6 and 7). In a second stage, the framework was validated by an additional two automotive OEMs (Chapter 8). The results of the validation phase are being presented at Chapters 5, 6, 7 and 8. Finally, the research limitations, conclusions, and recommendations are discussed in Chapter 9.

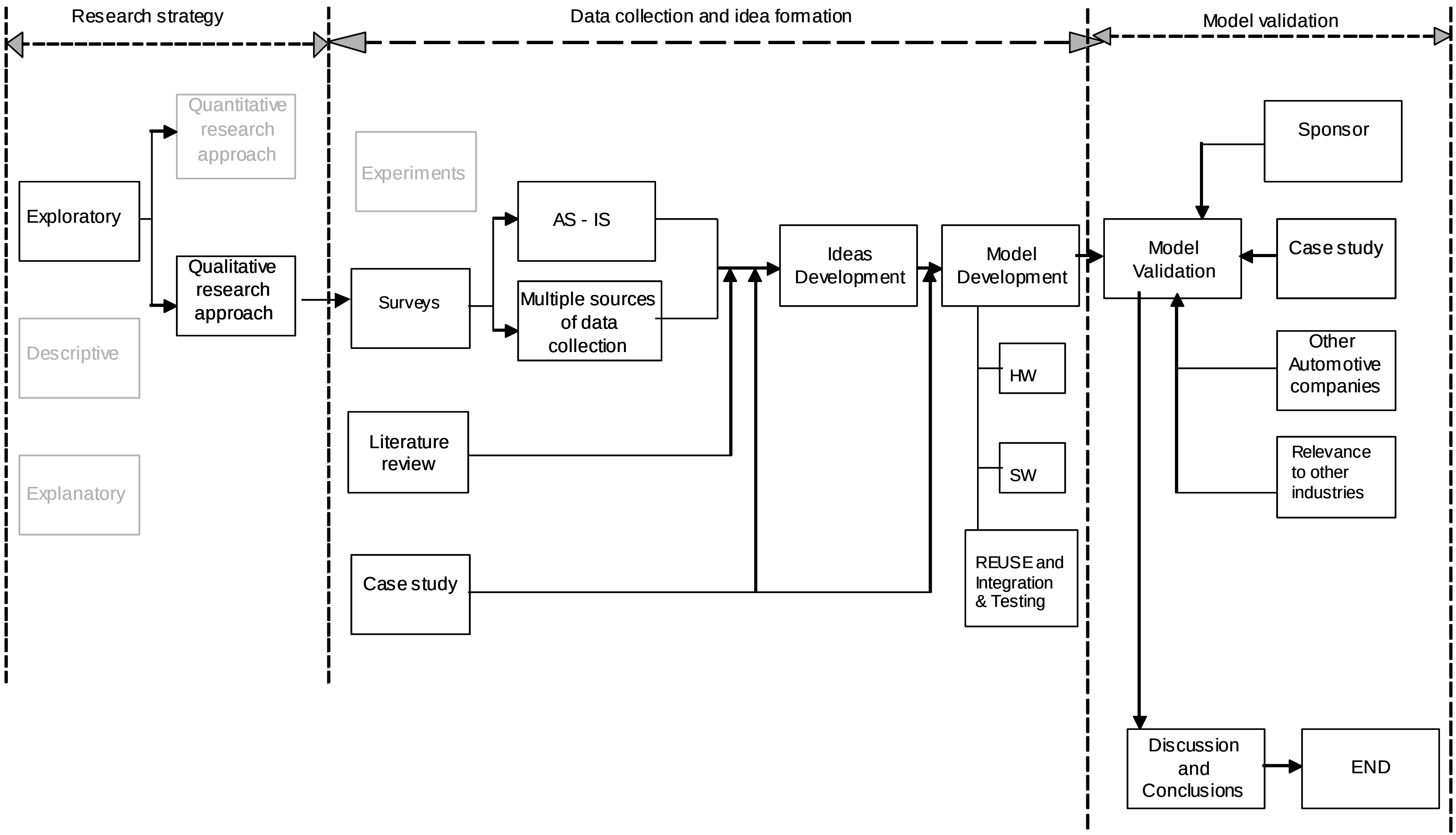


Figure 3.1: Research Methodology Map

3.5. Summary

In section 3.1, the research's aims and objectives were set. In order to accomplish the objectives of the research, an appropriate research strategy had to be designed. In section 3.2, available research purposes and approaches were reviewed. An exploratory research purpose was defined and a qualitative research approach was chosen (due to the exploratory nature of the research).

In addition, the survey and case studies strategies about conducting the research were chosen. The survey approach was chosen in order to get insight on the current situation and identify problems, opportunities and ideas that could support the model development. The case study approach was chosen mainly because the research is industrially sponsored, the cases were available and they served both in the stage of the model development as well as in its validation stage.

In section 3.3, the issues related to case study are analysed and the methods to deal with risks to research validity are also presented and the final research methodology is illustrated as a 'research methodology map'. In the following chapter, the findings of the survey regarding the current status (AS-IS) on ES D&D cost estimating in industry are presented. The structured approaches to data collection, such as process modelling and participant observation, and data, information, and other relevant issues regarding the survey environment are presented.

4. AS-IS model capture

In the previous Chapter, the research objectives were defined, and a research methodology presented. Within this chapter, the author discusses the initial survey (AS-IS) conducted with the use of questionnaires, semi-structured interviews and workshops. This Chapter details an AS-IS study for estimating the cost of an electronic system. In total, two automotive companies were investigated. In order to investigate if there are commonalities, differences and special cases across industries and various companies, the author expanded this survey to two companies of the aerospace/defense sector.

In section 4.1 the author describes the methodology used in order to conduct the study. A questionnaire was developed to support the interviews conducted with the cost estimating experts. Section 4.2 presents the results of the study regarding ES D&D cost estimating firstly within automotive, secondly within aerospace/defense and at the end, between automotive and aerospace/defense industries, comparatively. Section 4.3 presents general comments observed through out the study. Finally, in Section 4.4, Chapter summary and key points are presented.

4.1. Design of AS-IS study

4.1.1. Target Audience

The aim of this survey was to capture the electronic items cost estimating process followed in industry. Since the research's focus is on automotive sector, the researchers (Nikos Giannopoulos and Aitor Sarasua-Echeverria) utilising the contacts available to the University and from the industrial sponsors of the E-Mode project, contacted a number of automotive companies in order to investigate if they would like to participate in the AS-IS capturing process. Companies were contacted by phone or by email, they were introduced to the E-Mode project and they were asked if they would like to participate. Although various automotive companies were contacted, only one company (Company 2) accepted.

In addition to automotive sector, the researchers decided to expand their AS-IS model capture in other industries as well, in order to identify best practices and similarities and differences within the electronic items cost estimating processes followed across industries. Again, various companies from the aerospace/defence and consumer electronics industries were contacted, however only two companies from the aerospace/defence sector, Company 3 and Company 4 accepted to participate.

Two interviews and 3 workshops were conducted for this study, involving experts of varying knowledge and experience. The number of people interviewed per company varied from between one and three individuals. In the case of the first automotive company, two interviews with the electronics cost estimator were held: the first in order for him to present to the researcher the electronic items cost estimating process and the second one in order to develop, along side with the researchers the detailed AS-IS model based on the $IDEF_0$ (Integrated DEFinition Function Modelling 0) model ($IDEF_0$ is being presented in Appendix M). In the case of the remaining 3 companies, workshops with 2 to 3 participants were held. These experts, although of various skills (ie product engineer, purchaser, etc), they were selected from the interviewed organisations to be present in the workshops alongside with the cost estimator in order to provide a more in-depth view of the estimating process, since their expertise contributes on the estimating process. A complete list with the participants' job roles, and years of experience can be found in the next table:

Table 4.1: Workshops and Interviews participants

Title	Industry	Experience	Interview/ Workshop
Electronic Parts Cost Estimator (Finance)	Company 1	5 years	Interview
Manager, Cost Estimating	Company 2	15 years	Workshop
Manager, Electrical Product Cost Estimating and Control	Company 2	10 years	Workshop
Product Cost Estimating/ Project Estimator	Company 2	3 years	Workshop
Manager, Cost Estimating Business Cost Forecasting and Pricing	Company 3	10 years	Workshop
Head of Electronics Design	Company 3	10 years	Workshop
Manager, Electrical Product Cost Estimating and Control	Company 4	7 years	Workshop
Head of Electronics Design	Company 4	5 years	Workshop

4.1.2. Questionnaire Development

A semi-structured questionnaire was developed by the E-Mode team (Mr. Nikos Giannopoulos and Mr. Aitor Sarasua-Echeverria) in order to serve as a guide to both the interviews and the workshops. The questionnaire (which can be found at Appendix A) contains a set of questions that aimed to elicit the knowledge of the participating experts on how the electronic items cost estimating process is carried out, but not in a restricted way; it served as a 'guide' of keeping the conversation 'on-track', to allow the participants to expand their views and for the researchers, to make follow-up questions based on the received answers, as the experts often raised issues that are not covered by the questionnaire but might be also of significant importance for the researchers in order to have an overall picture of cost estimating process.

So, the survey's objectives are:

1. Identifying current practices on electronic items cost estimation in automotive as well as in other industries (aerospace/defence).
2. Capture and represent, in graphical form with the use of $IDEF_0$ standard the industrial collaborator's electronic items cost estimating process, in order to identify any bottlenecks, treads and opportunities
3. To compare electronic items cost estimating processes across industries, in order to identify common problems and potential opportunities for the electronic items cost estimating processes process enhancement.
4. To identify if any particular procedure is followed by any of the industries of any sector to assess the electronic items D&D cost.

4.1.3. Conducting the Interviews

As it was described earlier (section 4.1.2), the researchers followed a semi-structured interview approach. This enabled the researchers to define the depth of the answers provided by the experts and at the same time offered the experts the opportunity to expand their answers and reveal issues that were not covered by the questionnaire. All interviews and workshops were conducted by both members of the E-Mode research team (Nikos Giannopoulos and Aitor Sarasua-Echeverria). Each of the researchers performed his own analysis of results, since each researcher's interest lies in different research domains. The analysis of Mr. Aitor Sarasua-Echeverria covered only the PCB design and development side, whereas the analysis of Mr. Nikos Giannopoulos, which is presented in this thesis, covers the whole area of the design and development of embedded systems: SW and HW D&D, SW and HW Reuse and SW/HW integration and testing.

Interviews and workshops were both performed on-site at each company's location. All the interviews and workshops have been captured apart from paper in tape as well. This was done to ensure accuracy on interpretation and analysis of results, as well as a point of reference when a doubt occurs.

Capturing Electronic Items Cost Estimating process at the first automotive organisation (Company 1)

Capturing electronic items cost estimating process within the environment of the first automotive organisation involved two stages: first, to capture the process and in a second stage to represent this process graphically in order to identify any bottlenecks, threads and opportunities. The first stage was covered by using the semi-structured interview process described earlier. The second stage was covered by firstly making an introduction to $IDEF_0$ modelling process and then, working step by step alongside the electronic parts cost estimator to develop the graphical representations of each step of the process. It has to be stated here that $IDEF_0$ model was produced only for the first automotive organisation due to the lack of availability of experts to develop a detailed study with the remaining companies. However, the process of estimating the cost of developing an electronic item as well as any challenges faced was captured for these companies.

Capturing Electronic Items Cost Estimating process in the remaining organisations

Each of the workshops lasted (approximately) 2 hours, and the procedure followed in these workshops was that initially, the researchers presented an introduction to the E-Mode project, its aim and its objectives. After this initial stage, the workshop was initiated with the use of the semi-structured questionnaire. The benefit of the semi-structured approach was that offered the researchers the opportunity to follow-up on some of the interviewees' answers and reveal additional information, information that added value to the research and that the researchers wouldn't be aware of otherwise. At the end of the workshop, there was an open discussion about the E-Mode project and the electronics in general.

4.2. Analysis of AS-IS Study

4.2.1. Performing the analysis of results

Having the information collected, the results were analysed by examining the answers given in the questionnaire as well as by analysing the discussions that took place during the workshops. The researchers looked for consensus between participants' answers, as well as for commonalities, differences and unique to each company practices. The background of the interviewees was taken into account, and the taped information was used together with the documented information, to ensure documented information's accuracy and therefore validate them.

An overview of the key-issues identified during the workshops/interviews is being presented in the next table (Table 4.2). There were some differences identified between the companies of the automotive sector and there were some similarities identified within the companies of the aerospace/defence sector. Section 4.2.2 presents the AS-IS study regarding electronic parts D&D cost estimation within the first automotive organisation in graphical form using the $IDEF_0$ standard. Section 4.2.3 presents a comparison between automotive industries, section 4.2.4 presents a comparison between the aerospace/defence companies, and finally, section 4.2.5 presents a comparison between automotive and aerospace/defence companies. At the end, at section 4.2.6 general observations and key issues as they arose from this study are being presented.

Table 4.2: Summary of experts' answers to key questions (Cont.)

Questions	Company 1	Company 2	Company 3	Company 4
What are the cases which an electronic items cost estimation is produced for?	To assess the soundness of a supplier's quotation	From when the idea of a new car is conceived until management's approval for going on with its actual production, from management's approval until producing the first car, and from the first car to the end of the residual life of the car	To assess the soundness of a supplier's quotation	To assess the soundness of a supplier's quotation
Could you please explain the fundamental stages you go through when you develop an electronic items cost estimation?	The manufacturing cost is being estimated (materials, labour, process cost), then a number of mark-up costs are being calculated on top of it. The sum of manufacturing cost plus the mark-ups gives the final piece price.	Materials cost, labour, overheads, losses from prototype to production, profit, delivery and transportation are being summed-up to derive the piece price	Three-points estimates are introduced against supplier's quotations, and an iterative process is run, aiming to exclude risks in each of the iterations, until the item comes close to the 'target cost' they have internally set for the item under investigation.	It is done through the use of parametric models, developed in-house. The supplier's quotations are judged against the 'target cost' set by Company 4 based on its internal past projects database
During the electronic items cost estimation process, is there any input received by any other company department? If yes, what other departments are involved?	Sometimes there is an input from the engineers responsible for the item which is being estimated	By marketing, purchasing and engineering	Engineering (design engineers, requirements engineers, manufacturing engineers) and the project manager	No
How is all this information linked?	The electronic parts cost estimator is being supported by the engineer who is responsible for the item which is being estimated	The estimating process is being performed by a team consisting by the estimator, the buyer and the commodity engineer	The estimating process is being performed by a team consisting by the estimator and the engineers (design engineers, requirements engineers, manufacturing engineers) and the project manager	----------
Could you please outline the problems you are currently facing during the electronic items cost estimation process?	The estimating process is based on the estimator's and on the expert's experience, since the necessary information is being protected by supplier's IP. There is also no link between the item's specification and the cost being asked by the supplier. There is a big difficulty on assessing the D&D cost in general and the SW D&D cost in particular	There is no access to necessary information since they are protected by supplier's IP. Difficulty on assessing the SW D&D cost, even by using commercial packages.	There is no access to necessary information since they are protected by supplier's IP.	There is no access to necessary information since they are protected by supplier's IP.

Table 4.2: Summary of experts' answers to key questions (Cont.)

How do you assess D&D cost?	As a mark-up cost on top of the manufacturing cost	As part of the overheads. However, if an item is produced in house, then there is a formula to estimate it, derived by their own internal experience.	Through an in-house developed process.	Based on past projects, and on their experience and expertiece
How do you assess the HW D&D cost?	By examining the item and creating a BOM based on the best of their knowledge.	By examining the item based on their knowledge.	Based on their experience	Based on past projects, and on their experience and expertiece
How do you assess the SW D&D cost?	As a part of the D&D mark-up cost	Part of the overheads, they are now however experimenting with a commercial package. They also rely on the experience of people that have knowledge on writing embedded SW	They also rely on the experience of people that have knowledge on writing embedded SW	Based on past projects, and on their experience and expertiece
What additional data, information, method could improve the quality of the electronic items cost estimation process?	Access to data that are now inaccessible due to their IPR protection and a link between specifications and final cost.	Access to data that are now inaccessible due to their IPR protection and a link between specifications and final cost.	Access to data that are now inaccessible due to their IPR protection	Access to data that are now inaccessible due to their IPR protection

4.2.2. Embedded Systems Cost Estimating in the first automotive Organisation

A model serves as a system's representation, and it describes what a system does, what controls it, what it is working on, what means it uses to perform its functions, and what are its deliverables. A model is developed to provide system's better understanding in order to identify any problems in the seamless operation of the system and provide the basis for improvements, changes, or replacements. Additionally, a thorough understanding of a system provides a sound basis for comparison and benchmarking against other systems, in order for the best practices to be identified.

Modelling a process is the first step on decomposing a complex problem in smaller, more manageable sections, using a standardised mean. By representing a process in a graphical form, makes it easier for the process to be comprehended, understood and analysed. It also serves a basis of understanding if something is done wrong and should be changed.

Many process modelling techniques are available, with the most well known ones being:

- SADT (Structured Analysis and Design Technique) approach
- DFD (Data Flow Diagram) approach
- SSADM (Structured Systems Analysis and Design Methodology) approach
- IDEFØ (Integrated DEFinition Function Modelling 0) approach

The need to represent in a graphic format the embedded systems cost estimating process within the first automotive organisation led the researcher to investigate the IDEF family of models in order to decide which variation of this standard best matches our graphical representation requirements. $IDEF_0$ is one of the most used techniques used for model representation. It is a method designed to model the decisions, actions, and activities of an organisation or system. The need to graphically represent the process map and the fact that it is easy to use and popular in manufacturing industry was the motivation to adopt $IDEF_0$. This standard is presented briefly in the next paragraph.

Both $IDEF_0$ and $IDEF_3$ were considered since both are process description capture methods. The difference between the two is that $IDEF_3$ also considers the timing associated with each represented activity. However, for the scope of this research we are only interested with describing the activities involved, and not the time frames these activities take place or how much time they consume. In addition, IDEF syntax is well documented and supported and it consists a widely used process modelling technique. So, for the purpose of this research $IDEF_0$ was finally chosen.

The IDEF family of models is shown in Appendix M. $IDEF_0$ allows the user to identify any core activities and decompose the process under investigation. (Colquhoun and Baines, 1991). It is a standard widely used in industry since it is easy to use and understand. The syntax of $IDEF_0$ is illustrated in Figure 4.1, which is composed of these elements:

- **Activity Box** → It represents a concrete action to do.
- **Mechanism** → These ones are the elements which perform the activity.
- **Input** → Inputs are used by Mechanisms to make the activity.
- **Control** → Controls determine how and when the activity is done.
- **Output**→ Outputs are the result of the activity.

Inputs, Controls, Outputs, and Mechanisms are usually referred as ICOMs.

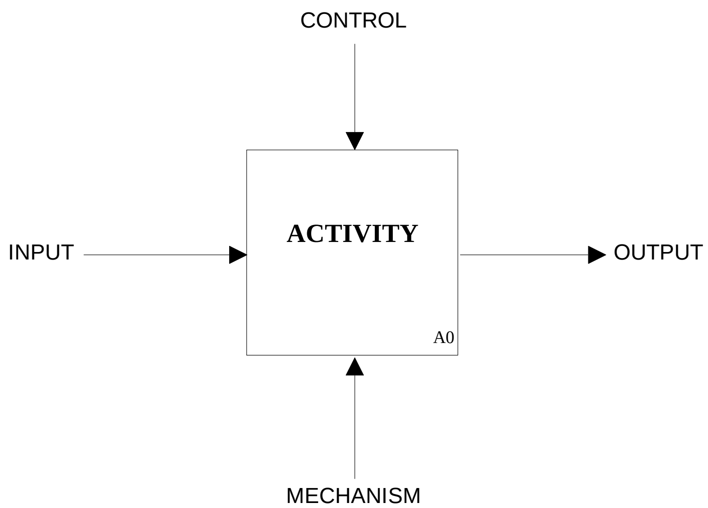


Figure 4.1: $IDEF_0$ Syntax Representation (taken from $IDEF_x$ standard)

Each activity could have more than one ICOMs associated with it. The first step on the modelling process is to define the ICOMs for the high-level activity of the process to be modelled. In the next step, the high-level activity is further decomposed in lower level sub-activities and the primal ICOMs are further refined, associated with each of the lower level sub-activities. This way, a hierarchy of diagrams is created:

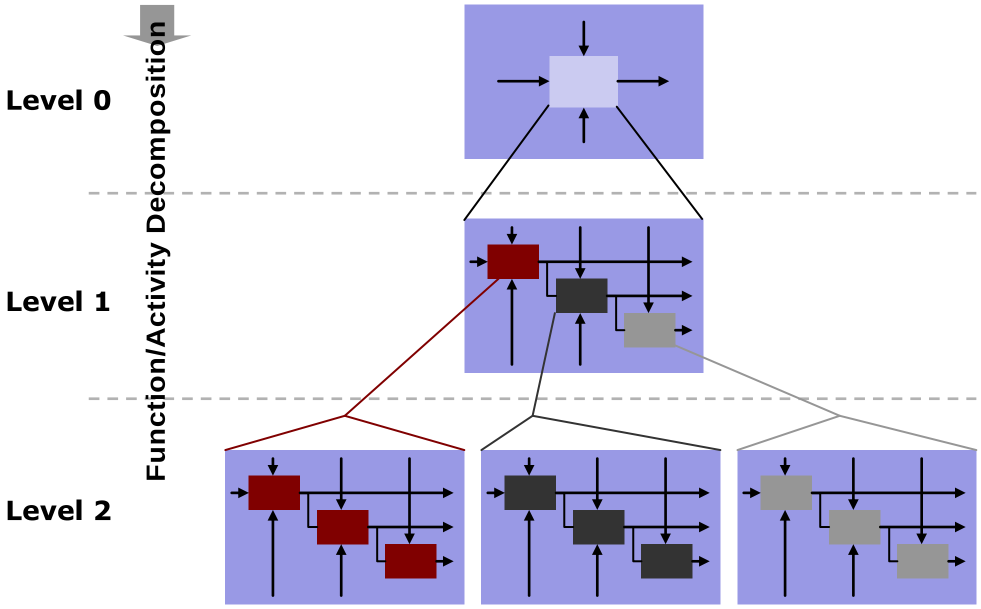


Figure 4.2: Function Decomposition Using $IDEF_0$ Notation (Rush, 2002)

Electronic parts cost estimating process was presented and explained to the researcher by the electronic parts cost estimator of the Cost Estimating department. At the beginning of the meeting, the estimator was introduced to the $IDEF_0$ standard and the way it works. After that introduction, the researcher and the estimator worked together on deriving the major nodes of the $IDEF_0$ model and then, modelling each one of these nodes. At the end of this process, the researcher brought all the nodes together and created the complete process' $IDEF_0$ graphical representation.

The electronic items cost estimating procedure is illustrated in the following figure (figure 4.3):

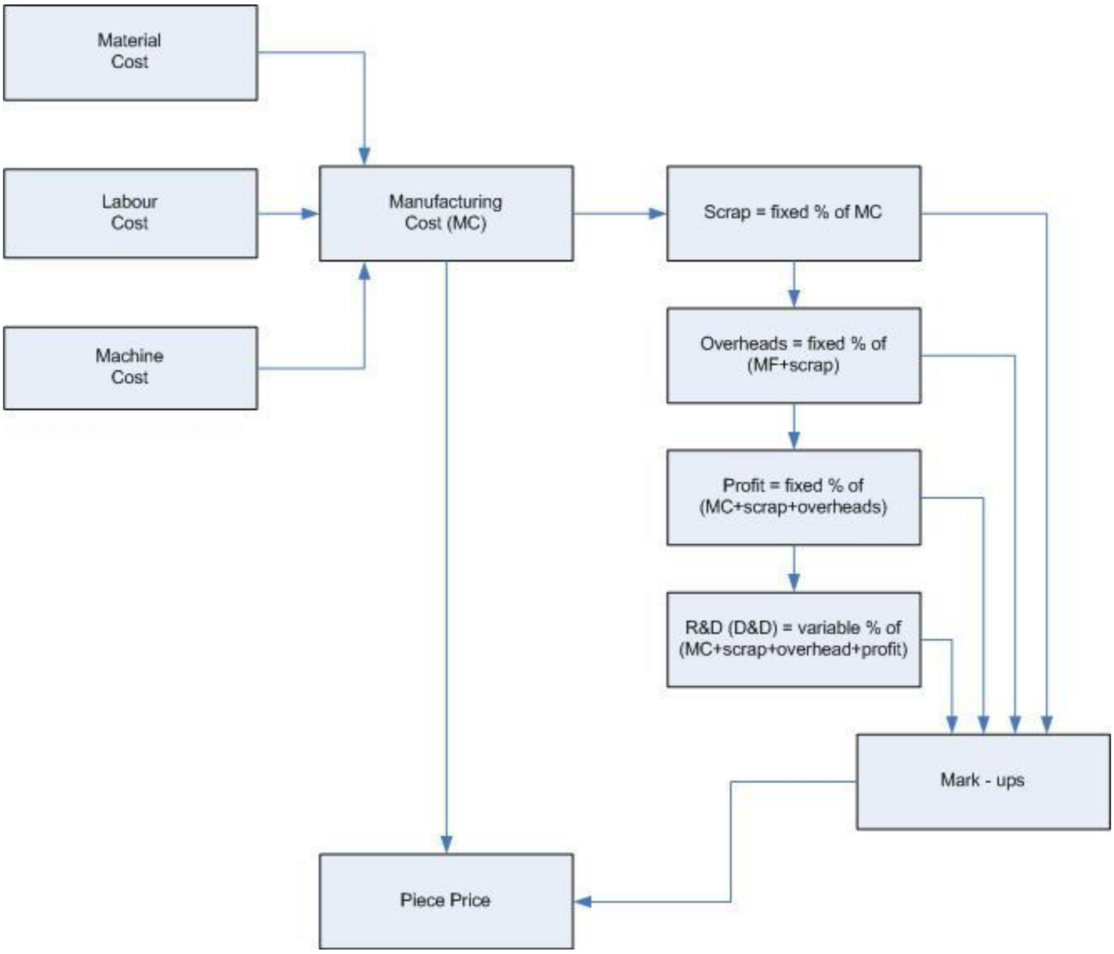


Figure 4.3: 1st automotive company ES cost estimating process

Therefore, regarding the AS-IS modeling in $IDEF_0$, 3 steps were identified:

- Manufacturing cost estimation
- Mark-ups cost estimation
- Piece price estimation

The relation between them is displayed in the following figure:

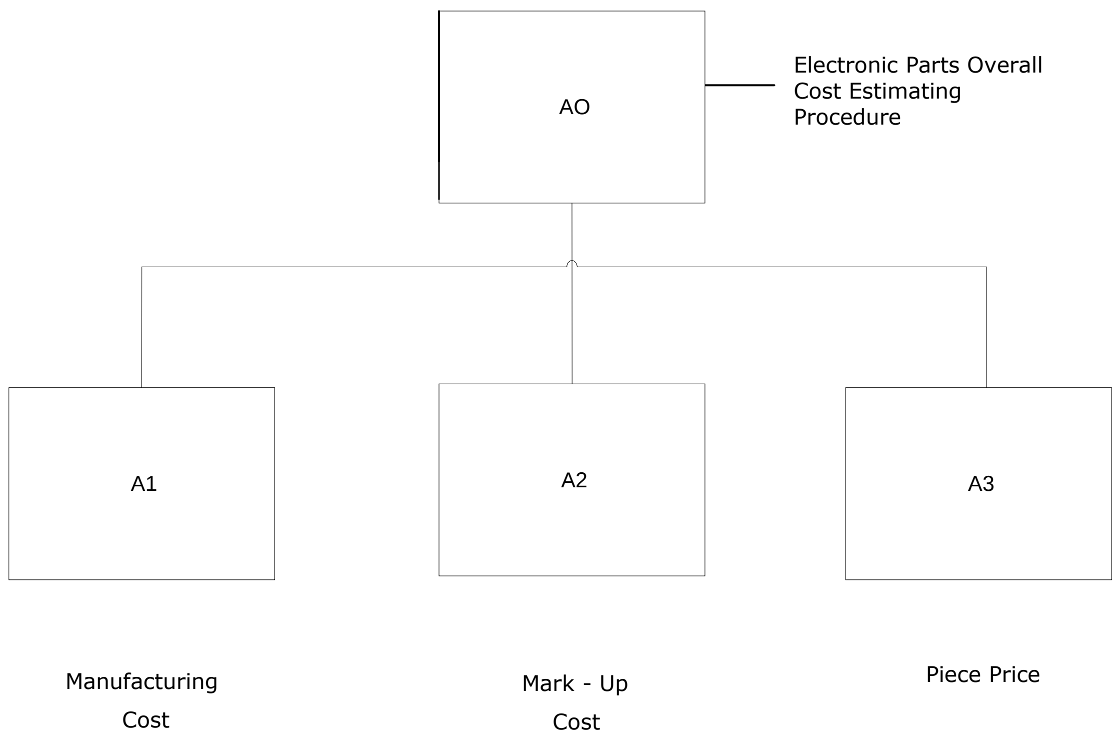


Figure 4.4: Relationships between individual IDEF0 Diagrams

The internal structure of each of the stages is as follows:

A0 – Electronic Parts Overall Cost Estimating Procedure

A1 – Manufacturing cost estimation

A11 – Material cost estimation
A12 – Labour cost estimation
A13 – Process cost estimation

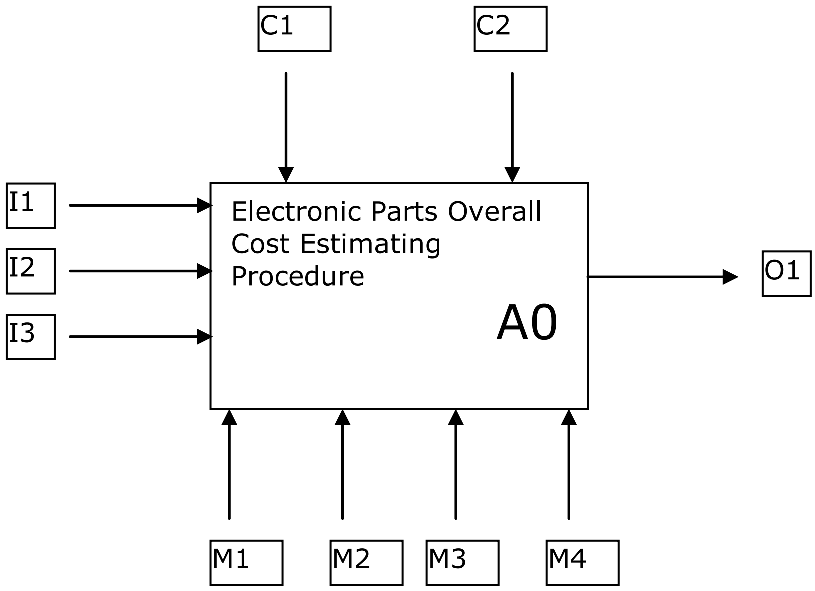
A2 – Mark ups cost estimation

A21 – Scrap cost estimation
A22 – SG&A cost estimation
A23 – Profit estimation
A24 – EDT cost estimation

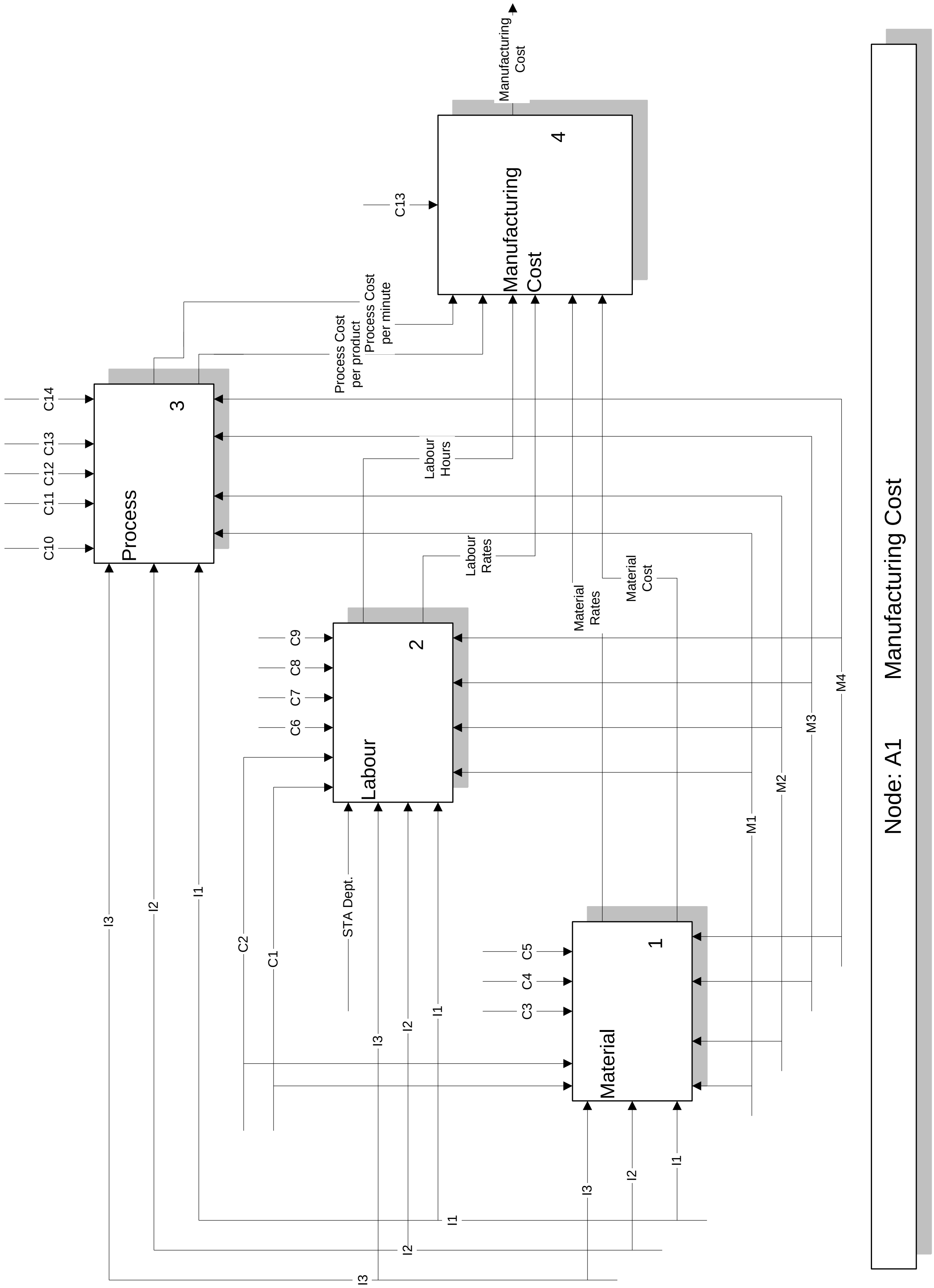
A3 – Piece price estimation

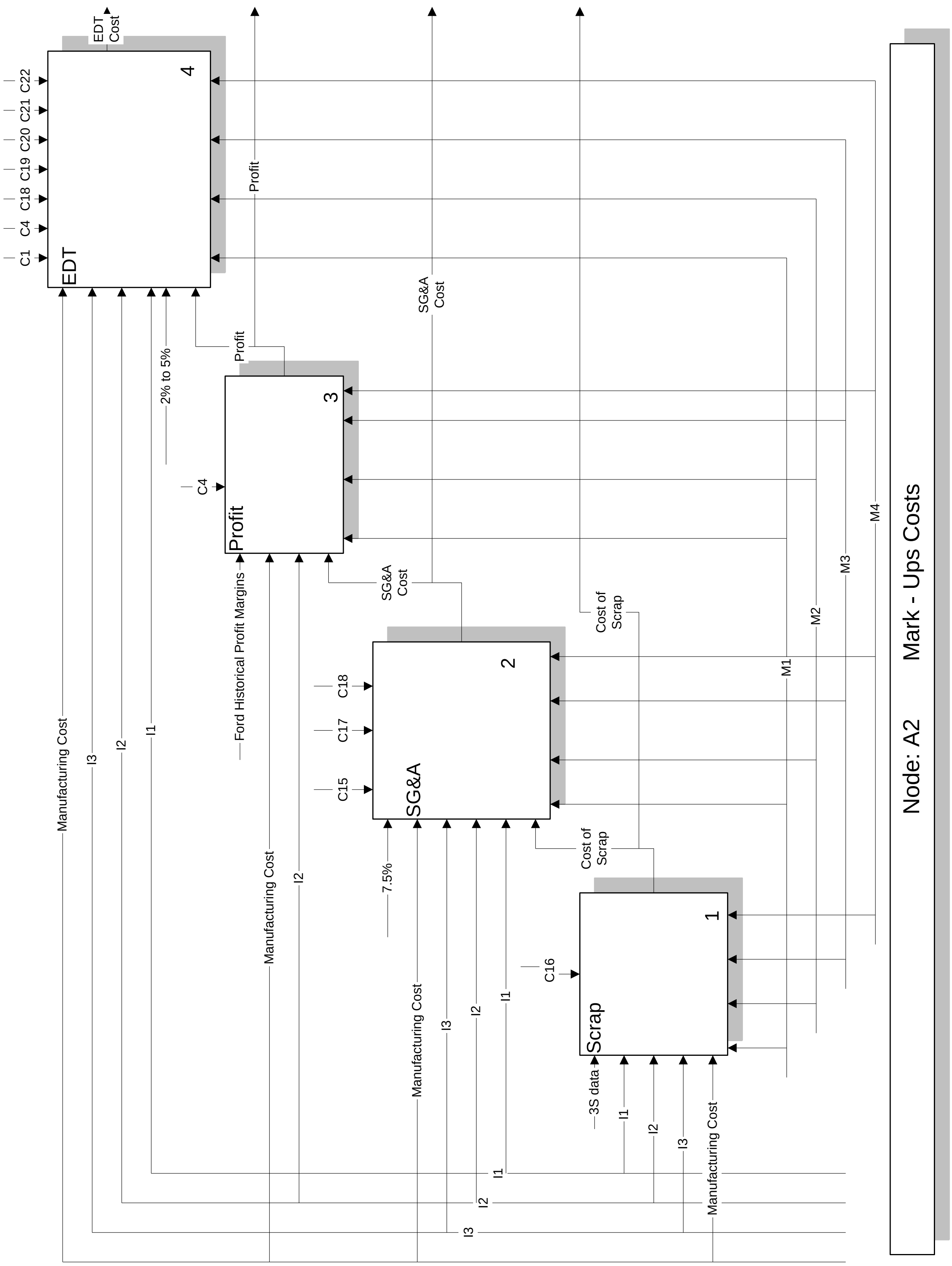
The individual $IDEF_0$ diagrams for each sub-system are presented in the next pages. Table 4.1 presents the list of Inputs, Outputs, Mechanism and Constraints identified and mentioned throughout the $IDEF_0$ diagrams. These diagrams have been validated by the interviewee in order to assure that the described processes have been captured and presented accurately.

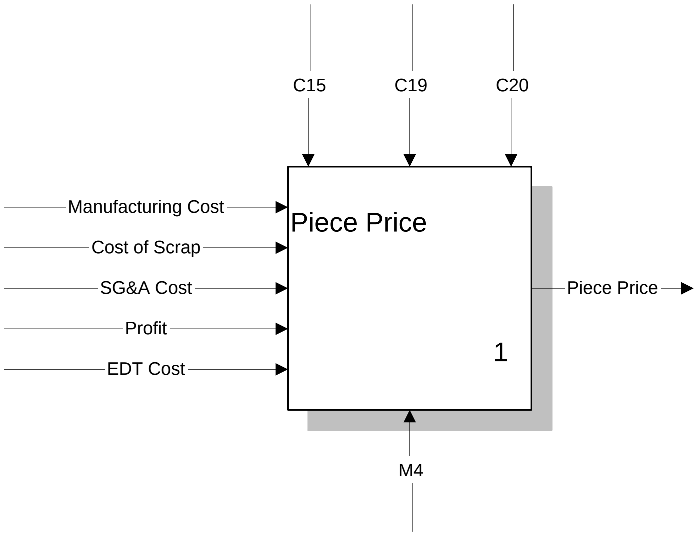
Memorandum:
Inputs:
I1: Ford databases
I2: Experience
I3: Supplier data
Constraints:
C1: Lack of Data
C2: Accuracy of data
C3: Special Items (Chips)
C4: New Technology
C5: People Knowledge
C6: Hours the operator spends on the machine
C7: Operator skill's level
C8: Level of Automation
C9: Number of shifts
C10: Availability (Up-Time)
C11: Depreciation method
C12: Maintenance and Repair
C13: Floor Cost, Energy and Oil Consumption
C14: Cycle time (seconds/product)
C15: Accuracy of previous calculations
C16: Process Difficulty
C17: Rule of Thumb
C18: Type of business
C19: Updated Technology
C20: Lack of Experience
C21: Efforts
C22: Testing Hours
Mechanisms:
M1: GRIMM
M2: CAPE
M3: Negotiations
M4: Excel tool
Outputs:
O1: Piece Price



Node O: $IDEF_0$ Costing Model


Material
1
Labour
2
Process
3
Manufacturing Cost
4
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
I1
I2
I3
M1
M2
M3
M4
STA Dept.
Material Rates
Material Cost
Labour Rates
Labour Hours
Process Cost per product
Process Cost per minute
Manufacturing Cost
Node: A1
Manufacturing Cost


Node: A2
Mark - Ups Costs
Scrap
1
SG&A
2
Profit
3
EDT
4
C1
C4
C18
C19
C20
C21
C22
EDT Cost
Profit
SG&A Cost
Cost of Scrap
2% to 5%
C4
Ford Historical Profit Margins
C15
C17
C18
7.5%
C16
3S data
Manufacturing Cost
I1
I2
I3
M1
M2
M3
M4



Node: A3 Piece Price Calculation

4.2.3. Presentation of results of the automotive companies

As it was stated earlier, two companies from the automotive sector (Company 1 and Company 2) were investigated in order for their electronic items cost estimating procedure to be captured. There were major differences identified between the two companies, which are presented bellow.

Company 1 outsources the whole production of its electronic parts, whereas Company 2 still produces a small number of them in-house. Both send a pack of specifications to the suppliers, describing the item to be manufactured. The difference though comes in that Company 2 breaks the complete system (car) in subsystems and sets 'target price' (and therefore cost) for each one of them, based on its experts' knowledge due to the in-house production. So, when a quotation from a supplier arrives, Company 2 is able to judge its accuracy and embark in further negotiations with the supplier.

A quite different procedure is followed by Company 1. Company 1 sends the requirements of the product to be manufactured to the supplier and ask for a quotation. However, Company 1 sets no 'target cost'. When the supplier's quotation arrives, then Purchasing and Engineering judge its accuracy; Cost Estimating is only sometimes called in this process.

The above highlight another difference between the two companies. Either it is the requirements elicitation and specification phase or judging or negotiating a quotation, Company 2 uses always a team of people to perform these tasks: an electronics cost estimator, an electronics components engineer and a purchaser. The rationale behind this approach is that by combining these skills together, a better assessment on a quotation can be derived by decreasing the uncertainty involved. This is not what was observed in Company 1.

Both companies do not differentiate HW and SW design and development cost estimation when estimating the cost of electronic systems; they consider an electronic item as a whole. In fact, for both companies D&D cost is part of the overheads. Company 2 however is experimenting with a freeway SW cost estimating package, which is still away from formal deployment due to the experts' unfamiliarity of the package's way of operations and of the metrics used. The electronic items cost estimating procedure for automotive company 2 is illustrated in the following figure (figure 4.5):

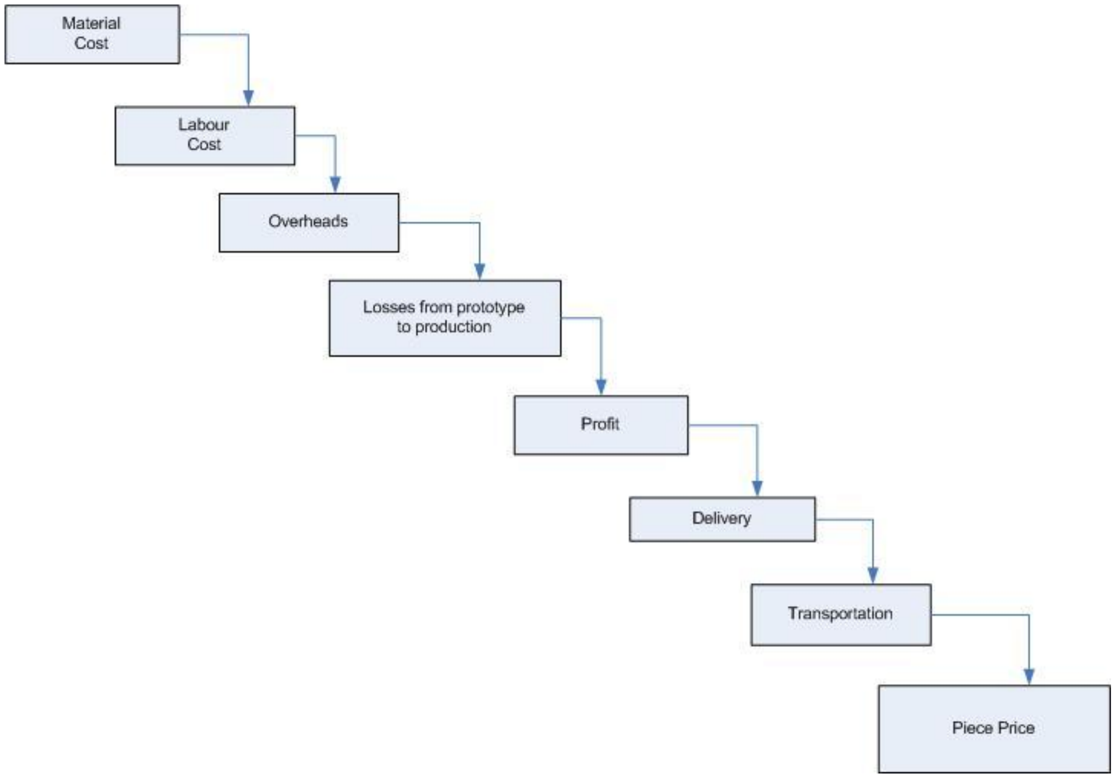


Figure 4.5: 2nd automotive company ES cost estimating process

4.2.4. Presentation of results of aerospace companies

Two companies from the aerospace sector (Company 3 and Company 4) were investigated in order for their electronic items cost estimating procedure to be captured. Company 4 outsources its whole production of electronic items, whereas Company 3 still produces a limited number of electronic items in-house.

However, both companies follow the approach of 'target cost'. They break down the system to be built in sub-systems and set a 'target cost' against each one of them.

The difference comes in that Company's 3 'target cost' is based on their experience due to their in-house production, whereas Company's 4 'target cost' comes from the in-house developed parametric effort estimation models. Because Company 4 produces a rather stable product (its product do not undergo significant changes through time, therefore even electronic items do not change considerably), Company 4 was able to create parametric models to estimate the design and development as well as the manufacturing cost of electronic items.

Company 3, either in the requirements elicitation and specification phase or when judging or negotiating a quotation, it always uses a team of a cost estimator, a manufacturing engineer and a design engineer. By concurrently engaging all these different skills of people decreases uncertainty, either it is the start of the project (requirements capture phase) or judging a quotation; there is no benefit delivering '’perfect’’ requirements for an item if manufacturing can not produce it, or having –costly- changes later in the production process due to unclear requirements.

Company 4 judges a quotation (including the D&D cost) based on their in-house developed parametric models, whereas Company 3 judges the D&D cost by developing, based on their in-house gained expertiece, 3 points estimate (minimum, maximum, most likely) to incorporate any risks or any assumptions for each of the sub-systems and for the whole system as well. The model is run several times; in each iteration some of the risks or the assumptions are taken out and the model is then re-run. This is continued until the model meets the 'target cost' set on the product.

In case of judging a supplier's quotation, the supplier's estimation is also incorporated on this model and the model and the procedure follows exactly the same way. The only difference is that if the supplier is perceived as 'good' (good history records and he provides 3-point estimates), then his 3 point-estimates are being directly incorporated on the model; however, if the supplier is perceived as 'bad' (no good history, provides 1 point estimates), then Company 3 'creates' the other two points on his estimates and then the three points are incorporated on the model. The embedded systems cost estimating procedures for the 2 aerospace companies are illustrated in the figures 4.6 and 4.7 bellow:

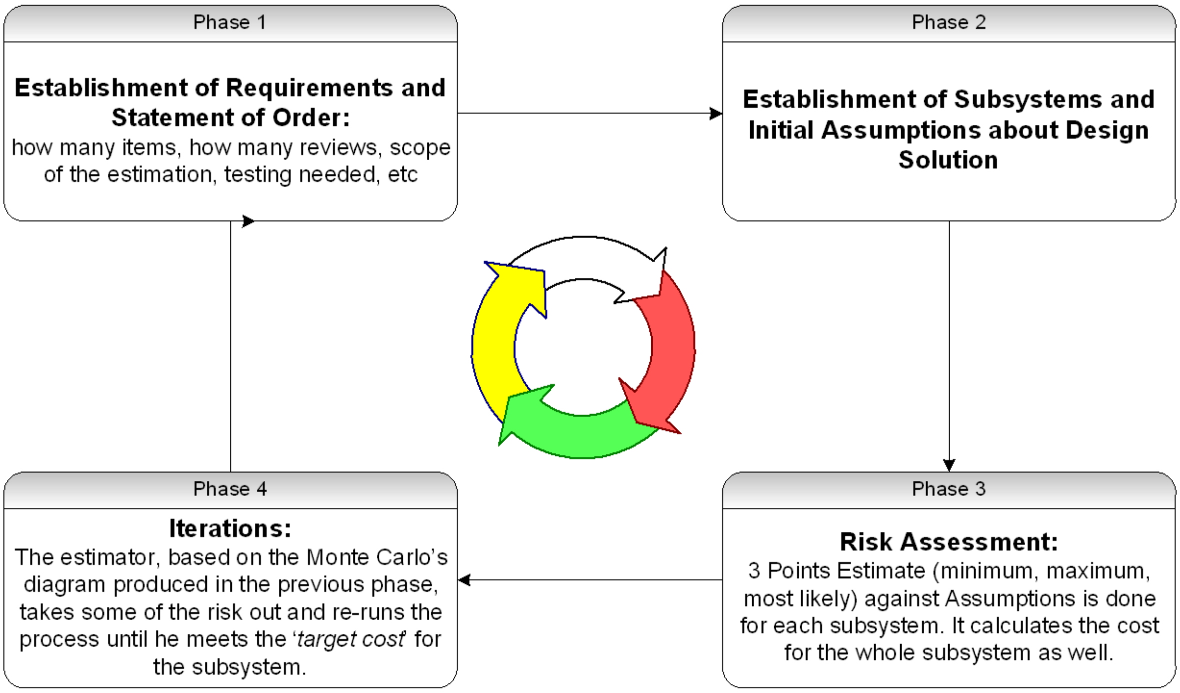


Figure 4.6: 1st Aerospace company (company 3) Electronic Items Cost Estimating procedure (Sarasua-Echeverria, 2003)

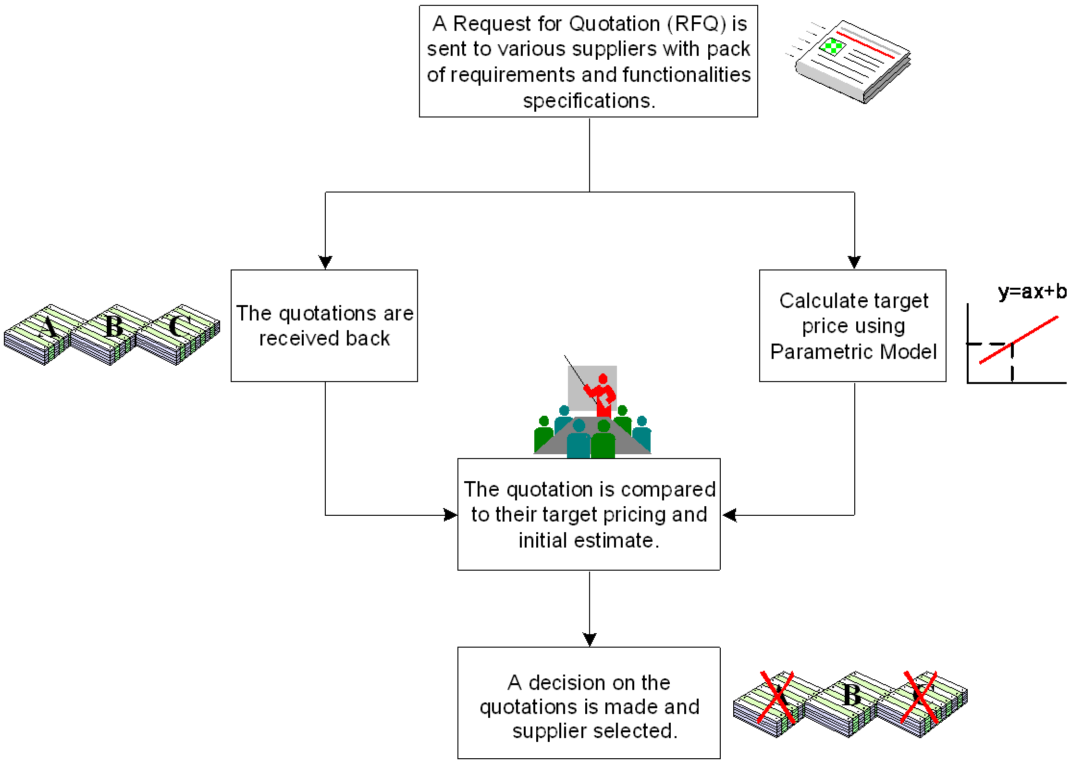


Figure 4.7: 2nd Aerospace company (company 4) Electronic Items Cost Estimating procedure (Sarasua-Echeverria, 2003)

4.2.5. Comparison of results of automotive and aerospace companies

After the electronic items cost estimation processes of the individual industries have been examined and the results have been presented, the next step is to compare electronic items cost estimating processes across industries, in order to identify common problems and potential opportunities for the electronic items cost estimating processes process enhancement.

Two major issues seem to be emerging as similarities: the first is that in both industries the commodity is broken down in subsystems and these subsystems are allocated a 'target cost', due to existing experience either from past projects from in-house production. This ensures better judgement on quotations and at the same time it allows for trade-offs in cost between subsystems according to their importance, keeping at the same time the overall product cost under control.

The second issue is that both the estimation process and the requirements capture/specification phases are being performed by a team of experts (a typical team usually includes an estimator, a R&D engineer, a manufacturing engineer responsible for the item and a purchaser). This ensures that requirements/specifications are 'set right at first time', avoiding costly revisions at a latter stage, as well as for better assessment of a quotation because of the existing knowledge/experience the team applies on the negotiations.

Apart from these similarities, there are also some differences observed. In automotive industry, due to the increasing need for improving car's efficiency, offering better value for money and reduced costs, changes in electronic items are often. However, this is not the case in the aerospace/defence industry, where the product is of a more stable nature.

In the automotive industry the price of the commodity is heavily affected by the competition in the market. That means that when in the past a car manufacturer was making a product (car) and then he was putting a profit on top of its cost to arrive at its price, nowadays the price of the car is affected by how much the competition in this car's range is charging. That means that the car manufacturer will have to reduce its cost close to the competition's, and that means that this cost reduction has to be reflected across the vehicle. This creates the need for the automotive industry to control costs to the limit. This does not appear to be the case in aerospace/defence industry, where the commodity is more of a stable nature and the competition is not as fearce.

Due to the long development product time and the small changes imposed due to the stable nature of the product, aerospace/defence industry has widely adopted and implemented data and past projects collection, especially to its own involvement in the design and development of electronics. This allows companies of that sector to develop their own cost estimating models. However, this is not the case in the automotive industry, where technology and products are changing with a very fast pace and therefore the creation of parametric models is not feasible.

For the automotive companies there is no model or methodology to estimate the D&D cost, whereas aerospace/defence industries can assess it through their own developed methodologies. All companies do not differentiate between HW and SW D&D effort estimation; they encounter the electronic item as one single product. Only Company 3 has an in-house developed methodology for estimating SW D&D effort estimation, which however could not be disclosed to the researchers. Company 3 was also the only one with a defined in-house procedure for assessing the D&D cost of an electronic item, although even this procedure is heavily based on the experts' knowledge and expertiece. For all the companies, it is easier to assess the manufacturing cost rather the D&D cost.

4.2.6. Results validation

The results of this survey are being presented in this chapter in a summarised form. The results of this survey, before they are presented here, were sent to the participating companies to examine them for their accuracy and therefore validate them. All participating companies were happy with the presented results, therefore the results of this survey are validated.

It has to be stated here that because of the limited number of companies that accepted to participate in this survey, the results tend to serve more as an indication rather than a definite outlook of the industries represented, and they are limited to the industries represented. Although participating experts agreed with the presented results, further investigation, to additional to automotive and defence/aerospace industries (consumer electronics, etc) would be necessary to produce a definite outcome.

4.2.7. General observations

Through the AS-IS analysis, the need for more accurate electronic parts cost estimation as well as for D&D (for HW, SW and the item as a whole) effort estimation became evident. Most of the companies have better understanding of the manufacturing cost, but lack knowledge to challenge the ES D&D cost. It was recognised by the most of the participants on this survey that the key to more accurate cost estimation is access to up-to-date information. The most common practice today, especially in the automotive sector, is that many estimators have their own 'databases', consisting of materials (brochures, catalogues, etc) which they have collected through their interaction with suppliers or their own research (ie: on Internet, conferences, etc). However, this information is not universal across the company and in addition, they are not usually up-to-date.

Most of the participants expressed the need for having access to a database with up-to-date information like overhead costs, material, labour costs, tooling etc. Along with this though, came the observations that developing and maintaining such a database would be an extremely difficult and time consuming task. It was also recognised the need for offering a way of assessing SW and HW D&D cost, since it becomes a growing issue for the companies.

Another important issue that was highlighted by all the participants was although the key to more accurate information is access to up-to-date information the issue is that this information is not available. The problem comes from the fact that most of the companies are outsourcing both the D&D and the production of their electronic items, and therefore all the information regarding D&D and manufacturing an electronic item is property of the supplier, and the company has no access to it. The situation becomes more complicated by the fact that even the supplier delivers that kind of information, is not always easy for the companies to assess the correctness of the information provided.

If suppliers do not wish to release information, then data from past projects have to be used. However, as it was stated earlier, only a few companies have created databases with data from past projects. Because of this unavailability of data, Expert Judgement is the most popular method for cost estimating. Therefore, there should be a structured and easy to use framework to link specifications to the cost estimating process, in order for the ''suppliers' block'' to be by-passed, especially for the D&D cost of an electronic item, since, as most of the participants stated, they are much more confident on estimating the cost of the manufacturing process.

Lastly, cost estimating an electronic item should be a process done by a team of people, not only a cost estimator. A team consisted by a component engineer, a purchaser and an estimator can produce much better results than the estimator alone, since it can also judge the technology behind the item, identify if the price offered is a good value for money and also make trade offs between associated systems. The same stands for D&D an electronic item, since a team of experts ensures that requirements/specifications are 'set right at first time', avoiding costly revisions at latter stages in the production process.

4.3. Key observations from the results.

From the analysis presented above, the following key observations can be derived:

- There is no standardised practice/method neither for electronic items cost estimating, nor for estimating their D&D cost. In the contrary, each company follows its own approach/method.
- Team estimating (estimator, component engineer, purchaser, etc) proves more effective than the estimator working alone, either it is D&D cost or manufacturing cost estimation.
- Accessing information necessary for estimating the cost (manufacturing or even D&D) of an electronic item proves most of the times difficult, since this information are most of the times property of the supplier and they are not disclosed.
- If information is available, they are either on the estimator's 'private' database or they come from a database. 'Private' databases are not universal throughout the organisation and both are not most of the times up-to-date or they do not cover all the information an estimator needs.
- Breaking down a system into sub-systems and setting a 'target cost' against each one of them seems to be the followed practice in industry.
- Some industries fail to collect past projects, which are necessary when other information is not available. In the absence of these data, the estimator has to use Expert Judgement, which is the most popular estimating method across industry. Even in case information is available, Expert Judgement is utilised by the estimator when performing his estimation.
- There is no differentiation between HW D&D and SW D&D when estimating the effort of D&D an electronic item; the SW part of the system is usually estimated as an overhead of the manufacturing cost.
- The difficulty of accessing information due to the variety of reasons described in the previous paragraphs creates the need for a framework to link the product specifications with the cost estimating process, by-passing this way the ''supplier's information blockage''.
- The method will need to be reusable so that cost engineers will be able to create estimates for D&D cost faster in the future for similar products.

The researcher believes that the development of this framework will provide a way for the estimators to estimate the D&D cost of an electronic item based on its specifications since in most of the cases that's the only information the estimator has access to.

4.4. Summary

In section 4 the author explained why this AS IS study was necessary. In Section 4.1 the proposed methodology was described, a questionnaire was developed and a semi-structure approach to the interviews was utilised. Section 4.2 the main analysis of the study is presented. The importance of a framework to link the specifications of an ES with its D&D cost was identified and established. Section 4.3 summarised some of the general findings across the different industries reviewed. Section 4.4 concludes with the final observation and main points on the finding as identified by this study.

In the following Chapter the development of the ES D&D cost estimating framework, based on the ES specifications, is being presented. The development of the framework is based on the analysis of real case studies (specifications of real ES) obtained by the first automotive organisation and provides a structured approach towards ES D&D cost estimation for the automotive industry.

5. TO – BE process Development

In the previous chapter (chapter 4) a detailed AS-IS study for estimating the cost of an electronic system was presented. The chapter concluded with the key observation that within automotive OEMs there is no model/framework for estimating the ES D&D effort; on the contrary, D&D effort is a standard percentage in the overheads.

This chapter describes the development of a TO-BE process that incorporates a step by step approach to D&D effort estimation within the ES cost estimation process. The TO-BE process is firstly developed and validated within the sponsoring organisation, whereas the ES D&D effort estimation framework was further validated by two other automotive OEMs.

5.1. Design of the TO-BE study: Capturing D&D Effort

In order to investigate the embedded systems D&D effort estimation problem further, the researcher designed and performed a workshop within the first automotive organisation (hereafter: the ''D&D effort capture workshop''). Having already identified that the ES D&D effort depends on HW, SW, InTres efforts as well as on Reuse, the reason for this visit was to run a workshop with experts in electronics in order to understand relative contribution of HW, SW, Reuse and Integration efforts to the overall ES D&D effort. The workshop result was complemented by the observations from literature review.

5.1.1. Target Audience

In order to accomplish that, the researcher organised a workshop in the sponsoring organisation's premises with four experts in the electronics domain. The objectives of this workshop were:

1. To drill further down in the ES D&D effort estimation domain and understand how experts perceive the issues of HW and SW D&D and the effects of Reuse and Integration towards the estimation of the overall D&D effort,
2. To identify potential areas of improvement, and
3. Based on (1) and (2), to produce the TO – BE model for the ES D&D effort estimation

Four engineers with experience in electronic parts design and development participated, and their opinions were collected using a semi-structured questionnaire. It has to be stated here that their experience and therefore their opinions reflect the experience gained within the automotive sector only and from past projects accomplished within the sponsoring organisation.

These experts were selected from the interviewed organisation as they were considered to be the people with the most experience and expertise on the electronics within the sponsoring organisation, and therefore they would provide an in-depth and detailed view of the situation. A complete list with the participants' names, job roles, and years of experience can be found in Table 5.1 bellow:

Table 5.1: Workshop participants

Name	**Job Role**	**Experience**
Engineer 1	**System Engineer,** Body and Security Electronics, Electronic Subsystems, EESE	8 years
Engineer 2	**System Engineer,** Security & Convenience, Electronic Subsystems, EESE	5 years
Engineer 3	**System Engineer,** Safety Electronics R&VT, Electronic Subsystems, EESE	5 years
Engineer 4	**System Engineer,** V34X Chassis Electronics, Electronic Subsystems, EESE	7 years

5.1.2. Questionnaire Development

A semi-structured questionnaire was developed by the researcher in order to serve as a guide on keeping the workshop on track. The questionnaire (which can be found in Appendix C) contains a set of questions that aimed to elicit the knowledge of the participating experts on SW and HW D&D and how Reuse and Integration affect D&D of electronic items, but not in a restricted way; it served as a 'guide' to keep the conversation 'on-track', to allow the participants to expand their views, and for the researcher to make follow-up questions based on the received answers. The experts often raised issues that are not covered by the questionnaire but are important to the researcher in order to have an overall view of the cost estimating process.

The questions asked were the following:

1. Please distribute 100% of electronics design and development effort between SW D&D, HW D&D and Integration and Testing (Intres), for an item D&D from scratch.
2. How do you currently assess HW D&D effort?
3. How do you assess HW complexity?
4. Can HW D&D be judged based on the item's specifications?
5. What are the problems with HW D&D effort estimation?
6. How do you currently assess SW D&D effort?
7. Can SW D&D be judged based on SW's specifications?
8. What are the problems with SW D&D effort estimation?
9. Which are the issues that affect reuse?
10. Can reuse be predicted?
11. Which are the issues that affect integration and testing?
12. Can integration and testing be predicted?
13. When Integration and Testing results are not satisfactory and the item has to be re-examined, what is the system's area that is affected the most?

These questions were chosen by the researcher, having in mind that their purpose was

- to help the researcher reveal how experts confront HW, SW, Reuse and Integration and Testing D&D effort estimation, and
- to help the researcher identify the factors that affect each stage (HW, SW, Reuse and Integration and Testing) of the ES D&D effort estimation process.

5.1.3. Conducting the Workshop

As it was described earlier (section 5.1.2), the researcher followed a semi-structured approach in conducting the workshop. This enabled the researcher to define the depth of the answers provided by the experts and at the same time offered the experts the opportunity to expand these answers and reveal issues that were not covered by the questionnaire. The workshop was performed on-site of the sponsoring organisation's location and lasted for 2 hours. All the information regarding the workshop has been captured in paper, to ensure accuracy of interpretation and analysis of results as well as a point of reference when a doubt occurs.

Capturing the experts' opinions involved the following procedure: initially, the researcher presented to the experts an introduction to the scope, the aim and the objectives of the workshop, in order to 'set the scene' and in a second stage the workshop was initiated with the use of the semi-structured questionnaire. The benefit of the semi-structured approach was that offered the researcher the opportunity to follow-up on some of the interviewees' answers and reveal additional information, information that added value to the research and that the researcher wouldn't be aware of otherwise. A summary of the experts' answers is presented bellow:

Table 5.2: A summary of the participants' observations (Cont.)

Questions	Participant 1 Engineer 1	Participant 2 Engineer 2	Participant 3 Engineer 3	Participant 4 Engineer 4
Please distribute 100% of electronics design and development effort between SW D&D, HW D&D and Integration and Testing (Intres), for an item D&D from scratch.	HW:15% SW: 35% integration of application SW with other SW (EFNOS, LANs, etc) Intres: 50% (integrating SW with HW, integration to car's electronic architecture and system testing)	HW: 15% SW: 30% (integration of application SW with other SW (EFNOS, LANs, etc) Intres: 55% (integrating SW with HW, integration to car's electronic architecture and system testing)	HW: 15% SW: 35% (application D&D) Intres: 50% (integrating application SW with other SW (EFNOS, LANs, etc), integration with HW, integration to car's electronic architecture and system testing)	HW: 10% (a lot of copy-paste from libraries) SW: 25% (application D&D and application's calibration) Intres: 65% (integrating application SW with other SW (EFNOS, LANs, etc), integration with HW, integration to car's electronic architecture and system testing)
How do you currently assess HW D&D effort?	Based on its implementation details (judging from the part itself) and supported by our own experience to assess its complexity.	Judging from the implementation and by using our experience to understand what it does and how it was manufactured	Based on our experience to understand its complexity	Based on our experience to understand how it works, plus assessing its complexity
How do you assess HW complexity?	Based on a list of factors based on experience (Memory type and size, number of components, number and type of interfaces).	Using some factors derived from experience (Functionality class, distributed functionality and number/type of components).	Based on the number and type of components, how distributed its functionality is and on what tests have to be taken.	Using a number of factors (type and size of memory, total number of components and how distributed its functionality is.
Can HW D&D be judged based on the item's specifications?	No, specs say 'what' the item should do, not 'how' it should do it. And this 'what' can be realised through a big number of alternative implementations.	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1
What are the problems with HW D&D effort estimation?	It always depends on the case. You can not judge based on the specifications, but you can not do it either by using the BOM, since there are parts (like ASICs) that we have no information on.	Specifications and BOM. Especially with using the BOM to ''compare'' two items, then you are deriving historical and not D&D cost.	Apart from the specifications problem, there is a problem by using the BOM to compare one item versus the other, because you can compare the standard items, but you can not compare parts like ASICs, that carry hidden intelligence by the supplier and they are protected by IP rights.	As the others.

Table 5.2: A summary of the participants' observations (Cont.)

How do you currently assess SW D&D effort?	As an overhead (fixed percentage) on top of the manufacturing cost	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1
Can SW D&D be judged based on SW's specifications?	No, specs say 'what' the item should do, not 'how' it should do it. And this 'what' can be realised through a big number of alternative ways.	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1
What are the problems with SW D&D effort estimation?	That the necessary information (ie code, flow diagrams, etc) is protected by IP rights.	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1
Which are the issues that affect reuse?	Even in situations we know there is reused involved, this is very difficult to be realised, since the item's details are protected by supplier's IP	The fact we do not have access to the SW or sometimes even in HW	The lack of information due to supplier's IP	The fact that most information on the item's implementation are hidden and protected by supplier's IP
Can reuse be predicted?	No, because there is no access to information. Even if we are sure there is any amount of reuse, we can not prove it.	The amount of reuse is something we have no access on. We can not predict it.	No, reuse can not currently be measured. We derive a percentage based on expert judgement but even then this is an indication, we can not prove it.	Reuse is hidden by the supplier. Even if we know it has been developed for someone else, the supplier says: 'I wasn't paid the D&D cost by the other company, so I now need my money back'. We do use expert judgement to derive an indication of reuse, but again, we can not prove it.
Which are the issues that affect integration and testing?	The SW and HW to be integrated and what will happen when the item is tested on real situations.	Depends on each case from the SW and HW to be integrated	On the SW and HW to be integrated plus calibration	The item's SW and HW, the side effects on the car and calibration.
Can integration and testing be predicted?	It's a case by case situation	It's case by case	It's case by case	It's case by case
When Integration and Testing results are not satisfactory and the item has to be re-examined, what is the system's area that is affected the most?	SW, because it is much easier to be modified	SW, because in this late stage it is not easy to redesign the HW	SW, because code can be modified but it is not easy to redevelop HW	SW because it can change easier than HW

5.2. Results

5.2.1. Performing the analysis of results

The results were analysed by examining the answers given as well as by analysing the discussions that took place during the workshops. The researcher looked for consensus between participants' answers, as well as for commonalities and differences. The background of the participants was also taken into account.

5.2.2. Results verification

The results from the workshop were sent back to the participating experts to examine their accuracy and therefore verify them. All participating experts were happy with the presented results, therefore the results of this survey are verified.

5.2.3. Presentation of results

The results from the workshop are summarised below:

- The distribution of effort when designing and developing an electronic item from scratch are as follows:

Table 5.3: D&D effort distribution according to engineers

	HW D&D	SW D&D	Integration and Testing (Intres)
1st engineer	10%	20% to 30% (average: 25%)	60% to 70% (average: 65%)
2nd engineer	15%	35%	50%
3rd engineer	15%	30%	55%
4th engineer	15%	35%	50%
Average:	13.75%	31.25%	55%

It has to be stated here that the percentages represented above are based on the experts' years of experience within the sponsoring organisation and their involvement with the supply chain. These percentages came very close to the ones observed in literature (paragraph 2.3). However, the researcher decided to adopt the percentages derived from the workshop (HW: 14%, SW: 31%, Integration and Testing: 55%) because they are targeted to the specific environment that is being investigated.

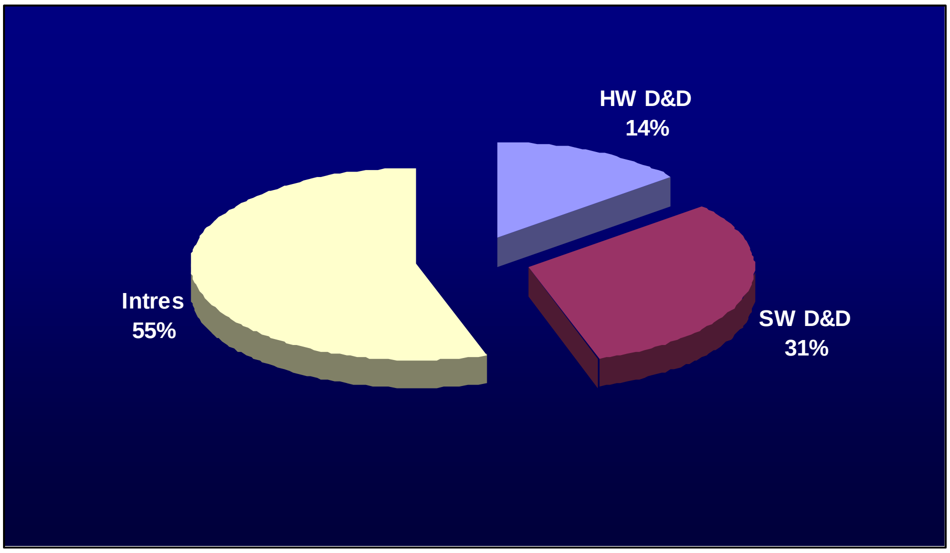


Figure 5.1: Distribution of electronic items D&D effort

Summarising the answers given by experts as they are presented in Table 5.2, the following key observations can be drawn:

1. HW and SW D&D effort estimation is done based on expert's knowledge, since necessary information is being protected by suppliers' IP rights.
2. Reuse (HW and/or SW) can not be estimated because (i) at the requirements or at the specification stage there are no detailed enough information to assist on that task, (ii) there is no access to this detailed information, since it is protected by suppliers' IPR, and (iii) even if they are sure that HW or SW has been reused, it is not possible to prove.
3. InTres is also very difficult to be estimated because it depends on the case at hand; it does not only depend on integrating the item's SW with its HW, it also depends on the unpredictable side-effects that occur when the item is integrated on the car's architecture. When the side impact actually happens, the item has to be taken out, examined, redesigned and retested. Usually, rework is applied to SW, since at that late stage of the development circle it is not easy to modify the HW.
4. The factors (Inputs, Outputs, Controls and Mechanisms) affecting each stage (HW, SW, Reuse and Integration and Testing) of the ES D&D effort estimation process have been identified:

    - HW Implementation Details
    - Expert Judgement
    - HW Complexity

- SW Specifications
- HW D&D Effort
- SW D&D Effort
- Integration Effort
- Past Data
- Lack of Data
- Expert Judgement
- New HW Technology

5.3. Development of the TO-BE model

In literature, it was concluded that (a) there is no framework/model for embedded systems D&D effort estimation, (b) none of the HW effort estimation methods are applicable in the ES domain, (c) for SW effort estimation: if the corresponding specifications are expressed in statecharts, there is no effort estimation method applicable to the ES domain, whereas if the specifications are expressed in UML Use Cases, the Use Case Points (UCP) method could be potentially used, and (d) there is no method/framework for estimating Reuse and Integration and Testing efforts.

As it was observed earlier in literature (paragraph 2.1) and during the effort capture workshop mentioned in the previous section, an ES consists of SW and HW, which are developed independently. Therefore, these 2 different development efforts should be taken into account when the overall ES D&D effort estimation has to be derived. In addition to SW and HW efforts, 2 additional issues should be considered: the issue of Reuse and the issue of Integration and Testing.

Reuse affects the D&D effort of both SW and HW because by reusing parts the developer is able to draw from similar past products and deliver the product quicker than developing the same product from scratch. The amount of Reuse applied depends on the situation at hand (paragraph 5.2.3). Integration and Testing is recognized as one of the major drivers in ES D&D, since it consumes around to 60% of the overall D&D effort (paragraph 2.5). This is because Integration and Testing happens at the end of the development cycle and therefore any problems discovered at that late stage require substantial amount of effort in order to rectify the problem.

From the analysis presented above, it can be concluded that a framework for estimating ES D&D effort depends on four different entities: HW, SW, Reuse and Intres. This also comes in accordance with the views engineers expressed in the effort capture workshop. These efforts should be combined in order to create the overall D&D effort.

Based on what has been presented in the previous 3 paragraphs, the researcher, using his own understanding, created and suggested an ES D&D effort estimation TO-BE model. Since ES D&D effort depends on the HW, SW, Reuse and Integration and Testing D&D efforts, the suggested by the researcher TO-BE model provides for estimation of these efforts, which are then aggregated to derive total ES D&D effort.. These developments are described in the following chapters (7 to 9). The logic of the suggested TO-BE model is presented, in a high-level view, in the following figure:

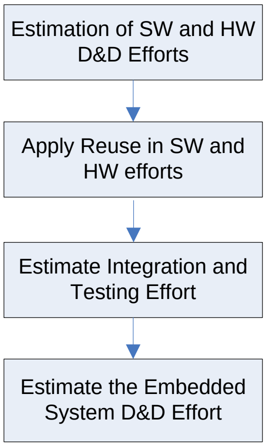


Figure 5.2: A step by step approach to ES D&D Effort Estimation

The overall D&D effort would be then estimated as follows: In the first step, HW and SW development efforts would be derived. Then, these two efforts are adjusted, amended by the appropriate percentage of Reuse applied. Intres can easily be estimated also: Since Intres is 55% of the overall D&D effort, once HW or SW effort has been derived, Intres effort can also be derived. Therefore:

Overall D&D effort=(HW D&D effort)*(Reuse percentage) +
(SW D&D effort)*(Reuse percentage) +
Integration and Testing D&D effort (eq. 5.1)

Based on the answers provided by the experts at the "effort capture workshop" (table 5.2) and utilising the factors that affect the ES D&D process (as they were also derived at the same workshop – paragraph 5.2.3), the researcher, using his own understanding, developed a detailed view of the ES D&D effort estimation

TO-BE model, using the $IDEF_0$ standard. The $IDEF_0$ diagram ICOMs, derived by the opinions of experts and literature review, are summarised at Table 5.4.

Table 5.4: TO – BE model's ICOMs

Inputs	I1	HW Implementation Details	Outputs	O1	HW D&D Effort
	I2	Expert Judgement		O2	SW D&D Effort
	I3	HW Complexity		O3	Amended HW or SW D&D Effort
	I4	SW Specifications		O4	Integration Effort
	I5	HW D&D Effort			
	I6	SW D&D Effort			
	I7	Integration Effort	Controls	C1	Lack of Data
				C2	Expert Judgement
Mechanisms	M1	Past Data		C3	New HW Technology

5.3.1. TO – BE model validation

The results of the TO – BE model development (Figure 5.2), before they are presented here were presented to the participating "effort capture workshop" experts for their practicality and usefulness. All participating experts were happy with the presented results, therefore the results of this development are considered initially validated. The model is further validated in future chapters through detailed development.

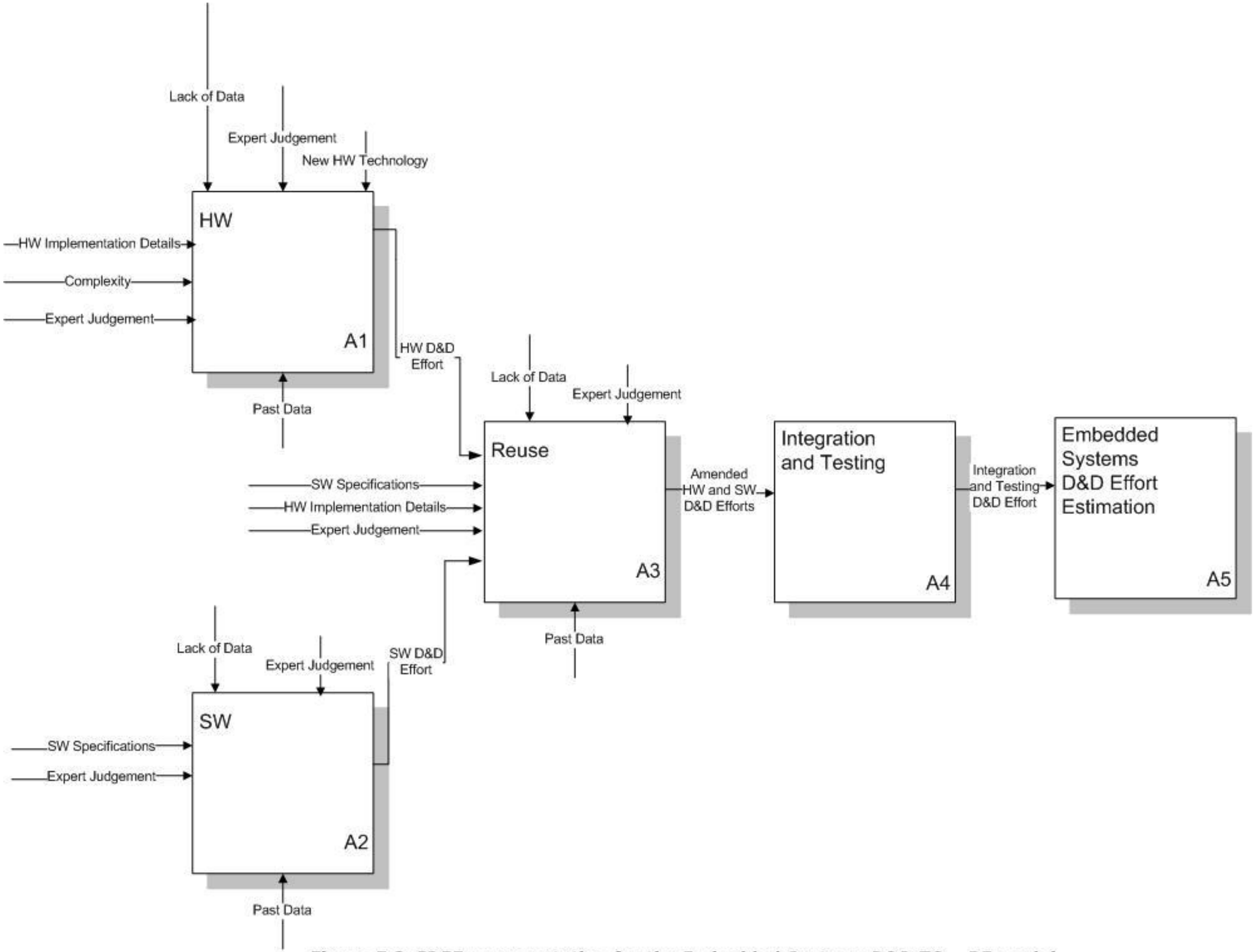


Figure 5.3: IDEF representation for the Embedded Systems D&D TO – BE model

5.4. AS – IS vs TO - BE

The TO – BE model presented here, offers a step by step approach for estimating the effort for ES D&D. It explicitely differentiates between HW and SW efforts and in addition it takes into account the Reuse and Integration and Testing effect towards the overall ES D&D effort estimation. The initial ES D&D effort estimation process, as this was presented in the AS-IS model, is highly subjective, since it is based on ad-hoc percentages applied by engineers. On the contrary, the suggested TO – BE model does not depend on ad-hoc percentages or information blocked by the supplier; the required information can be easily extracted by the OEM's specifications, and can be easily applied to the model, thus significantly limiting any bias or uncertainty.

The newly developed ES D&D effort estimation TO – BE model would be incorporated within the overall ES cost estimation process as shown in Figure 5.4 bellow. In pages 149 to 153 (Figures 5.5 a – d), the detailed view of how the TO-BE model will be incorporated within the sponsoring organisation's ES cost estimating process is being displayed:

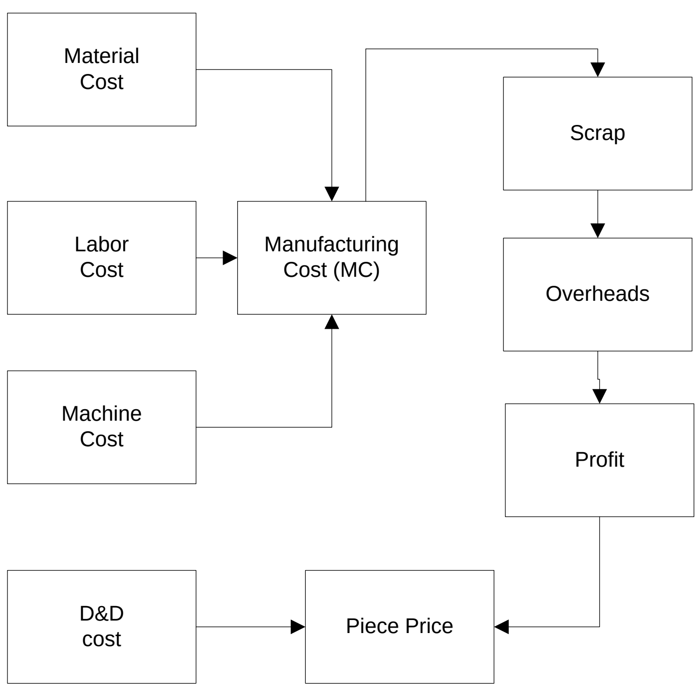


Figure 5.4: A step by step approach to ES Cost Estimation

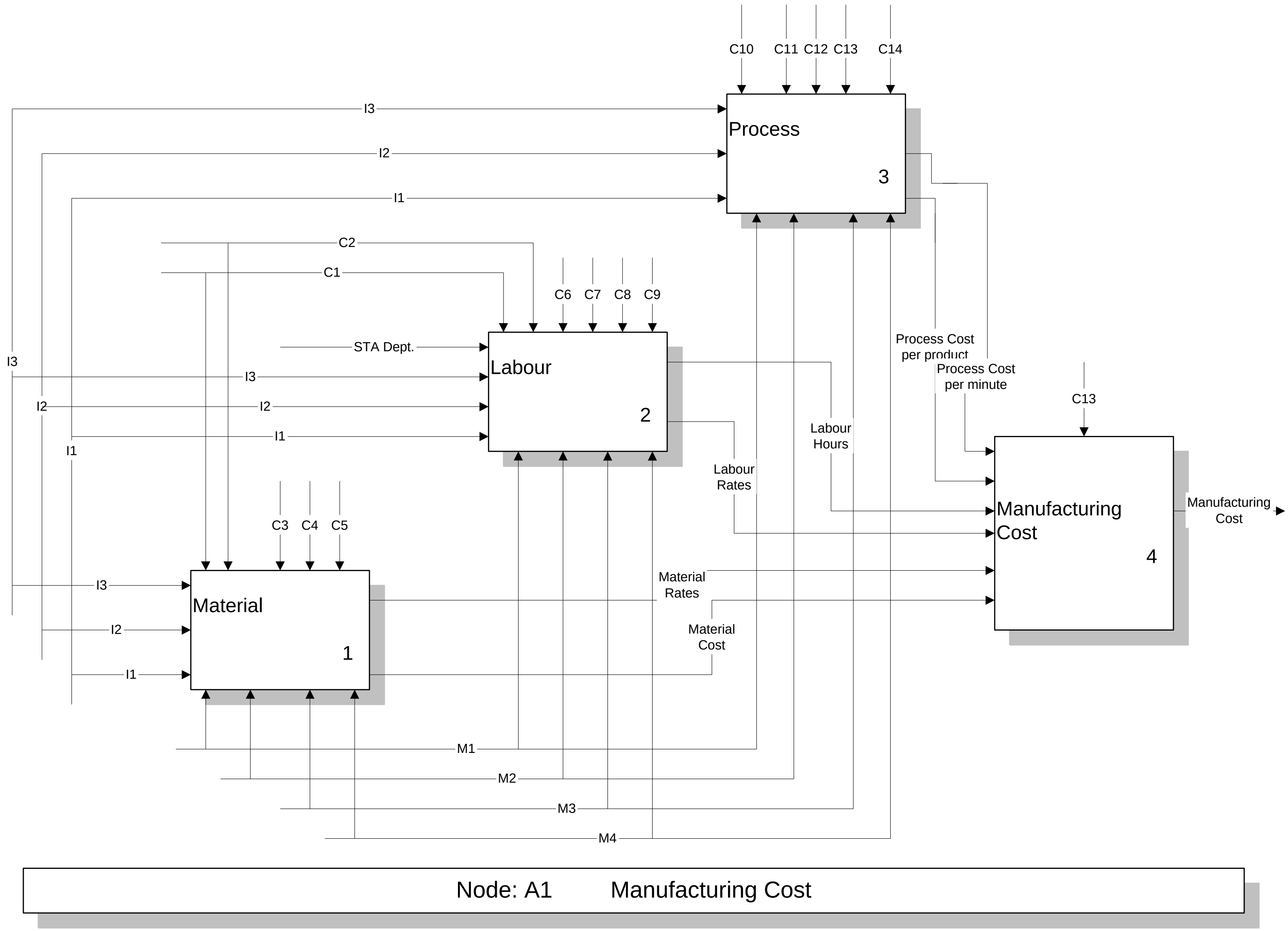


Figure 5.5a: TO-BE model ES Cost Estimation – Manufacturing Cost

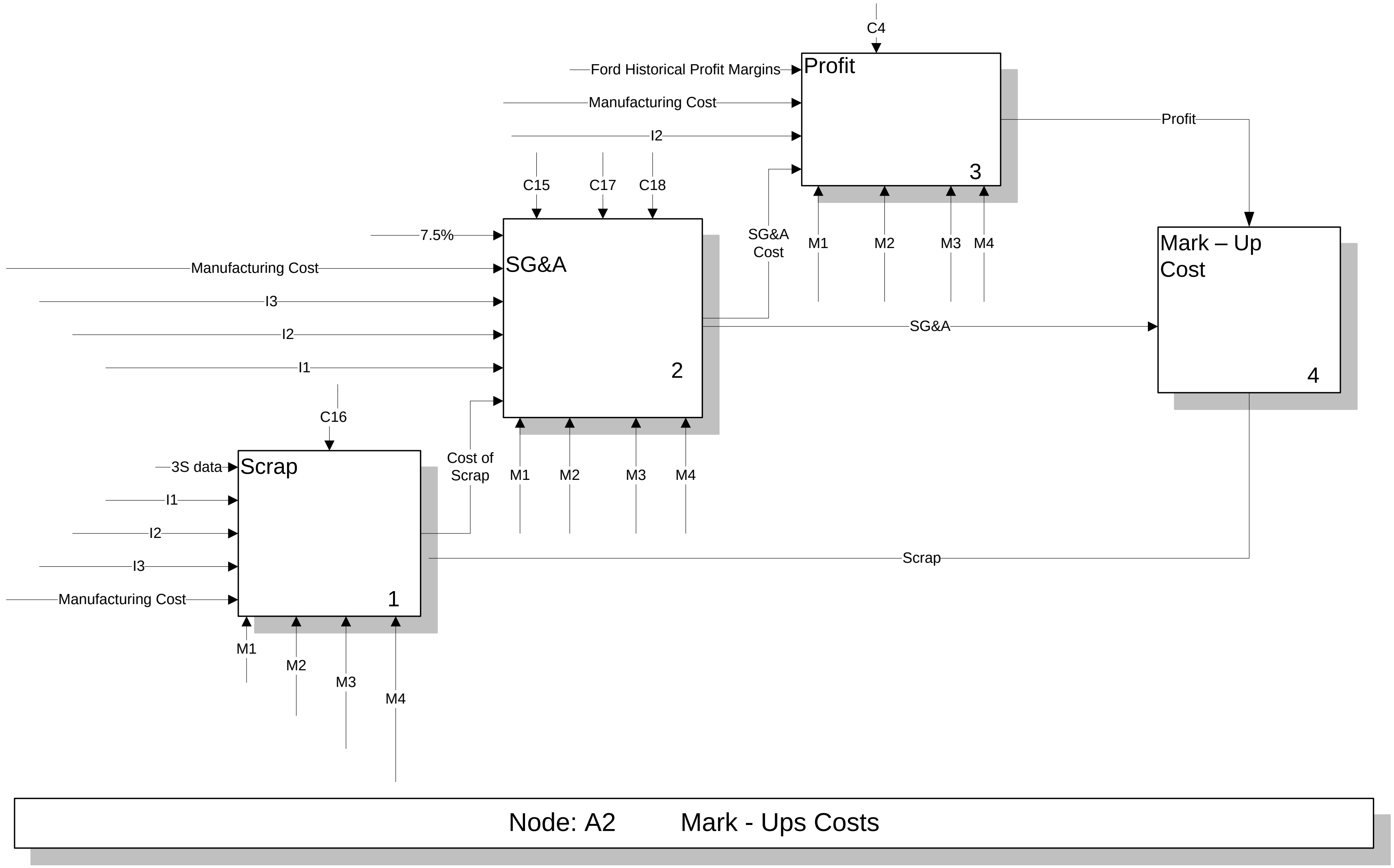


Figure 5.5b: TO–BE model ES Cost Estimation – Mark Up Costs

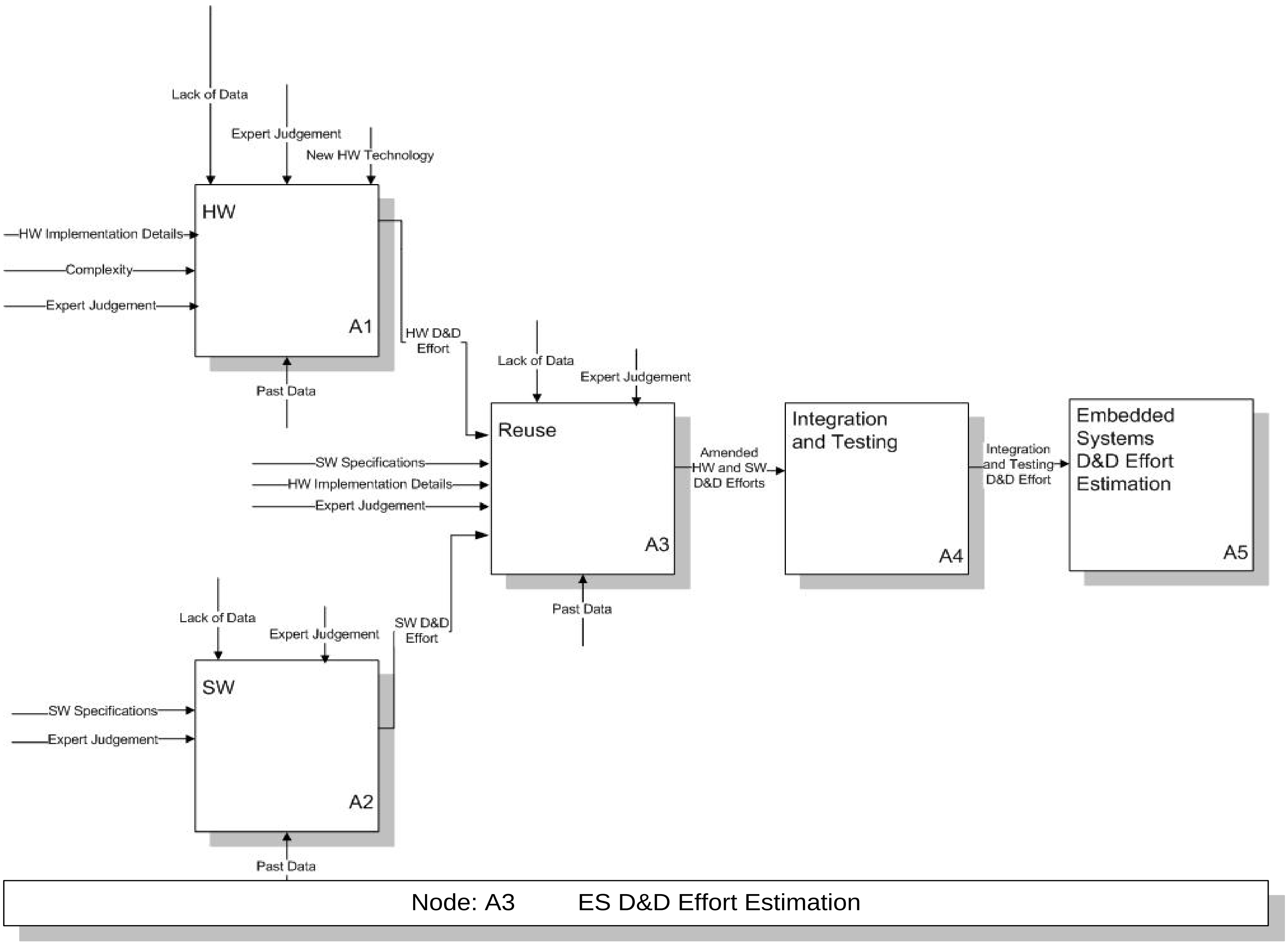


Figure 5.5c: TO-BE model ES Cost Estimation – D&D Cost Estimation

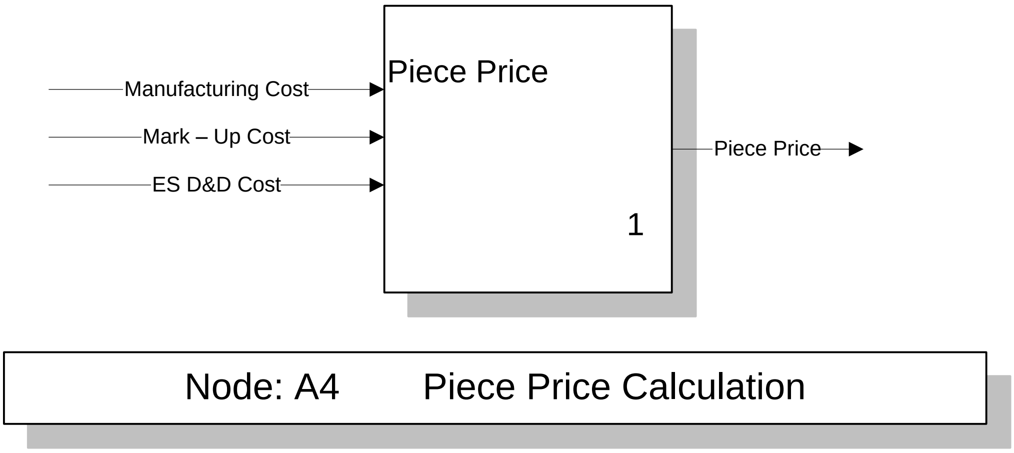


Figure 5.5d: TO-BE model ES Cost Estimation – Piece Price Estimation

5.5. Summary

In this chapter, a TO-BE model for estimating the D&D cost of an ES is presented. This TO-BE model explains the logic behind the developed (in the next chapters) D&D effort estimation framework. In Section 5.1 the design of the study, the questionnaire development and the workshop are described.

In Section 5.2, the analysis of the study is presented, where the differentiation between HW, SW, Reuse and InTres efforts is established. In addition, the relative contribution of HW, SW, Reuse and Integration efforts to the overall ES D&D effort is derived. In Section 5.3, the development of the TO-BE model follows.

Folowing chapters (6, 7, 8) present the development of the HW, SW, Reuse and Integration and Testing efforts estimation modules, which combine in chapter 9 in order to create the ES D&D effort estimation framework based on the TO-BE model presented in this chapter.

6. Developing a Cost Estimating Framework for Embedded Hardware Design and Development

In the D&D effort capture workshop, the engineers stated that in order to derive an estimation for the D&D effort of HW contained in an ES, they use their understanding to derive an overview of the HW's complexity based on a list of factors (''complexity factors'' hereafter) and then they use expert judgment in order for the D&D cost of the HW to be derived. However, this process is performed in an ad-hoc manner.

Therefore, if there could be a framework for linking the HW complexity with the HW D&D cost, the estimation process would become more transparent, structured and standardised. For that reason, the researcher decided to go forward on creating a framework for linking the complexity of the HW to be developed with the corresponding HW D&D cost, whose rationale is displayed in the following figure:

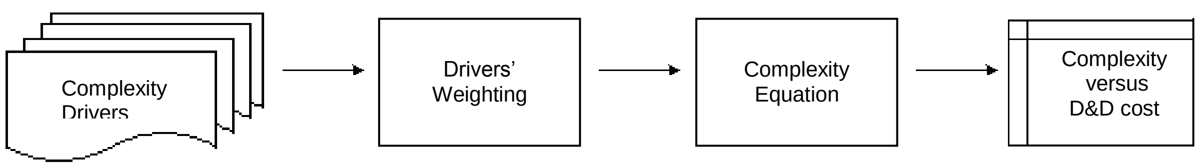


Figure 6.1: Linking Complexity to HW D&D Cost

In the first step, the drivers (the factors that affect complexity) are identified. In a second step, weighting for these factors is obtained, to account for each factor's significance towards the total complexity. Having derived both the factors and the factors' weightings, then a Complexity equation should be derived and values for complexity are obtained for ES HW where both a (supplier's or sponsoring organisation's proprietary) BOM and HW D&D cost exist. In the final step, complexity values are plotted against the corresponding HW D&D cost to obtain the ''Complexity versus D&D'' Cost Estimating Relationship (CER). By this way, when a new HW D&D cost has to be estimated, the engineer based on his experience would be able to derive a complexity value for this specific HW and from the CER to find the corresponding HW D&D cost.

In the following paragraphs the researcher presents the development of this CER as well as its validation by the experts. The framework was developed based on case studies from within the sponsoring organization. This is because of the difficulty in performing detailed cost estimating studies in other companies. However, as it will be sown in a latter chapter (chapter 9) the results were validated –apart from experts from the sponsoring organization- by experts from other automotive companies. That ensured generability of the results achieved within the automotive sector.

6.1. Developing the Complexity metric

The first step on developing the complexity metric is to identify the complexity drivers. In the D&D effort capture workshop, the experts stated the factors they consider as ‘’complexity indicators’’ when they apply expert judgement in order to estimate the D&D effort for the ES HW. The researcher complemented the experts’ answers and by doing this the following list of complexity drivers was derived:

1. Type of Components
2. Number of Components
3. Memory type
4. Memory Size
5. Number of Interfaces
6. Type of Interfaces
7. Functionality Class
8. Distributed Functionality
9. Test/Acceptance Criteria

After the complexity drivers have been identified, the next step on the complexity metric development is to obtain weights for these drivers, weights that express the importance of this complexity driver towards the HW D&D effort. However, as it was stated by the experts on the D&D effort workshop, the weight of each complexity driver towards the HW D&D effort is further affected by how complex this driver is internally, since there is ‘’fluctuating complexity’’ within each of the drivers. This is better explained by the means of an example, as this was given by one of the experts that participated in the D&D effort capture workshop:

‘’…The Number of Components is a Complexity driver for HW D&D. However, a HW implementation that has 10 Integrated Circuits (ICs) and no microprocessor (μP) does not bear the same complexity as a HW implementation with 10 ICs and 1 μP. The second HW implementation has a higher level of complexity…’’.

The same stands for each of the complexity drivers. For example, a HW implementation with 256kb of memory does not bear the same complexity with a HW implementation with a memory of 8kb. This means that since there is fluctuating complexity within each of the complexity drivers, adjustment factors should be introduced to account for the effect this internal complexity has in each complexity driver.

The researcher, based on his understanding and through the review of the literature, created the following table (table 6.4). The first two columns, ‘’Driver’’ and ‘’Name’’, describe the driver’s number and name. In the third column, ‘’Weight’’, the corresponding weight for this driver would be introduced. In the fourth column, ‘’Value’’, the researcher created, for each of the complexity drivers, a list of potential implementations and assigned corresponding ‘’adjustment factors’’ to each one of them to account for the fluctuating level of complexity within the drivers themselves.

The most complex implementation that could be encountered for each driver would carry on forward the complete complexity driver weight (therefore the adjustment factor of ‘’1’’ in this case). In any other case, the complexity driver’s weight that would be carried on forward for each one of the drivers would be the product of the complexity driver’s weight times the corresponding adjustment factor.

To illustrate the concept, let’s assume that the ‘’Type of Components’’ driver has a weight of 20%. If the HW to be D&D has more than 20 ICs and 2 microprocessors then the driver’s (‘’Type of Components’’) weight would be the complete driver’s weight, since this is the most complex situation –regarding this driver- that could be encountered. However, if the HW to be D&D has 1 to 10 ICs and no microprocessor, then the driver’s weight would be 20%x0.4.

Table 6.1: Initial Complexity Driver Table

Driver	Name	Weight	Value	Factor
d_1	*Type of Components*		1-10 ICs No micros	0,4
			1-20 ICs No micros	0,6
			1-20 ICs 1 micro	0,8
			More than 20 ICs 2 micros	1
d_2	*No. Components*		0 - 20	0,2
			21 - 50	0,4
			51 - 100	0,6
			101 - 300	0,8
			More than 300	1
d_3	*Memory type*		ROM	0,2
			PROM	0,3
			EPROM	0,4
			RAM	0,5
			OTP	0,8
			FLASH	1
d_4	*Memory Size*		8kb	0,3
			16kb	0,4
			32kb	0,5
			64kb	0,6
			128kb	0,7
			256kb	0,8
			512kb	1
d_5	*No. of Interfaces*		0 to 10	0,2
			10 to 20	0,5
			20 to 30	0,8
			More than 30	1
d_6	*Type of Interfaces*		Digital Slow	0,2
			Digital Quick	0,3
			Communication Bus (LAN, CAN, etc)	0,4
			Differential Inputs	0,6
			Ground Connections	0.2
			Combination of the above	1
d_7	*Functionality Class*		A (Radio)	0,3
			B (Immobiliser)	0,7
			C (ABS)	1
d_8	*Distributed Functionality*		70% distributed	0,3
			40% to 70% distributed	0,6
			Less than 40% distributed	1
d_9	*Test/Acceptance Crit.*		Normal requirements	0,1
			Special Mech. Requirements	0,2
			Special Temp. Requirements	0,4
			Special EMC Requirements	0,4
			Both Temp. and Mech. Requirements	0,6
			Both EMC and Mech. Requirements	0,6
			Both Temp. and EMC Requirements	0,8
			All of them	1

For any HW D&D case under investigation, using the above table, a complexity value could be obtained using the following formula:

$$C = d_1 \cdot r_1 + d_2 \cdot r_2 + d_3 \cdot r_3 + \ldots d_n \cdot r_n = \sum_1^n d_n \cdot r_n \quad \text{(eq. 5.1.)}$$

where C is the complexity value for that specific HW case study, d_n is the weighting of each driver, r_n is the driver's adjustment factor and n is the number of drivers.

6.2. Metric Validation and weight allocation

The researcher, in order to validate the derived metric and to obtain weightings for both the complexity drivers and the adjustment factors, interviewed 3 experts from within the sponsoring organisation. A list with the participants' job roles, and years of experience can be found in the following table:

Table 6.2: Survey participants

	Job Title	**Years of Experience**
Expert 1	**System Engineer,** Body and Security Electronics, Electronic Subsystems, EESE	8 years
Expert 2	**System Engineer,** Security & Convenience, Electronic Subsystems, EESE	5 years
Expert 3	**System Engineer,** V34X Chassis Electronics, Electronic Subsystems, EESE	7 years

These experts were selected from the sponsoring organisation since they were considered to be the people with the most experience and expertiece on the HW D&D domain, and therefore they would provide an in-depth and detailed view of the situation.

6.2.1. Questionnaire Development

A semi-structured questionnaire (found in appendix F) was developed by the researcher in order to serve as a guide on conducting the interviews. It contained a set of questions that aimed to assist experts on validating what was presented to them and on providing weights on the complexity and the adjustment factors. It also served as a 'guide' of keeping the conversation 'in-track', to allow the participants to expand their views, and for the researcher, to make follow-up questions based on the received answers.

So, the interviews' objectives were:

1. Validate the list of complexity drivers
2. Validate complexity driver's subcategories
3. Validate complexity driver's subcategories adjustment factors
4. Provide weightings for the complexity drivers

6.2.2. Conducting the Interviews

The interviews were carried out following the procedure described bellow: Initially, the questionnaire was sent to the experts by email, and after a few days the researcher condacted the experts and performed the interviews over the phone based on the semi-structured questionnaire that had already been emailed to them.

At first, the researcher presented to the expert an introduction to the scope, the aim and the objectives of the interview, in order to 'set the scene' and familiarise the experts with the procedure. In the second stage, interviews were initiated with the use of the semi-structured questionnaire. At the end of the workshop, there was an open discussion about the HW D&D metrics in general.

The experts' answers and comments have been captured on paper, for validation purposes and future references. A summary of the experts' answers is presented bellow:

Table 6.3: Weighting allocation and validation

<table>
<tr><th colspan="5">Weights</th><th colspan="2">Validation</th><th>Comments:</th></tr>
<tr><th>Driver</th><th>Name</th><th>Expert 1</th><th>Expert 2</th><th>Expert 3</th><th rowspan="2">Subcategories Validation</th><th rowspan="2">Subcategories Adjustment Factors Validation</th><td rowspan="11">No additional comment was received</td></tr>
<tr><td>d_1</td><td>Type of Components</td><td>20</td><td>20</td><td>20</td></tr>
<tr><td>d_2</td><td>Number of Components</td><td>10</td><td>10</td><td>10</td><td rowspan="8">Experts 1 and 2:

The 'Test - Acceptance Criteria' complexity factor should be redesigned to cover 'IP protection' (protection from contact with any foreign matter and/or water) and extreme temperature requirements (>85^{0}C).

Experts 1 and 3:

In the ''Memory Size'' factor, there should be also the subcategory of 1MB memory size included.</td><td rowspan="8">Experts 2 and 3:

In the ''Distributed Functionality'' factor, the percentages assigned to the subcategories should be the other way around: 70% and more distributed functionality should have an adjustment factor of 1, 40% to 70% an adjustment factor of 0,6 and for distributed functionality less than 40%, an adjustment factor of 0,3.</td></tr>
<tr><td>d_3</td><td>Memory type</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>d_4</td><td>Memory Size</td><td>5</td><td>10</td><td>10</td></tr>
<tr><td>d_5</td><td>Number of Interfaces</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>d_6</td><td>Type of Interfaces</td><td>20</td><td>15</td><td>20</td></tr>
<tr><td>d_7</td><td>Functionality Class</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>d_8</td><td>Distributed Functionality</td><td>15</td><td>10</td><td>10</td></tr>
<tr><td>d_9</td><td>Test/Acceptance Criteria</td><td>5</td><td>10</td><td>5</td></tr>
</table>

As it can be observed by the experts' answers (as they are presented in table 5.6 in the previous page), 3 amendments were asked:

(i) The redesign of complexity driver d_9 (Test/Acceptance criteria) to cover protection against any foreign matter and/or water and extreme temperature requirements (for temperature more than 85^0C). Protection provided on a system against any foreign matter and/or against water is specified by the appropriate ''IP classes'' as they are set by the DIN 40050/IEC 529 Standard (displayed in figure 5.3). Each type of protection is specified from the standard indicator ''IP'' and a two digit code. The first digit describes the degree of protection against any foreign matter and the second digit describes the protection against water.

It was decided by the researcher that protection against any foreign matter should be treated separately from protection against water. This was decided by the researcher on the basis that although an embedded system could be protected from contact with any foreign matter by its actual position on the car's architecture (it is hidden, for example, in the car's structure) this does not necessarily mean that it is, at the same time, protected against the water.

So, protection from any foreign matter and protection form water were treated separately. For the protection against contact with a foreign matter, the standard IP classes (from 0 to 6, as they are described in the second column of figure 5.3) in three entries with the corresponding adjustment factors were introduced. For protection against water, the researcher decided to introduce only one subcategory (Water Resistance) with an adjustment factor of 0.8. This was decided by the researcher on the basis that for an embedded system there should not be partial protection against water; an embedded system should either be water protected or not.

Finally, temperature requirements were confronted by two separate entries in complexity driver d_9 (Test/Acceptance Criteria): Temperature Requirements/Normal Temperature with an adjustment factor of 0.6 and Temperature Requirements/Extreme Temperature with an adjustment factor of 0.7.

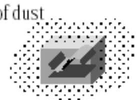
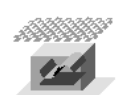
DEGREES OF CONTACT PREVENTION AND GUARDING AGAINST FOREIGN MATTER			DEGREES OF WATER PROTECTION		
First index digit	EXTENT OF PROTECTION		Second index digit	EXTENT OF PROTECTION	
	PROTECTION	EXPLANATION		PROTECTION	EXPLANATION
0	no protection	–	0	no protection	–
1	against large foreign bodies 	Protection of persons from accidental large-area direct contact with active or internal moving parts (e.g. hand contact), but no guard against intentional access to such parts. Protection of the object from access of solid foreign matter larger than 50 mm in diameter.	1	against water dripping vertically 	Water drops falling vertically must not have any harmful effect.
2	against medium-size foreign bodies 	Protection of persons from finger contact with active or internal moving parts. Protection of the object from access of solid foreign matter larger than 12 mm in diameter.	2	against water drippinig up to 15° 	Water drops falling at any angle up to 15° with the vertical must not have any harmful effect.
3	against small foreign bodies 	Protection of persons from touching active or internal moving parts with tools, wires or similar foreign bodies thicker than ø 2.5 mm. Protection of the object from access of solid foreign matter larger than 2.5 mm in diameter.	3	against spray water 	Water hitting the object at any angle up to 60° with the vertical must not have any harmful effect.
4	against granular foreign bodies 	Protection of persons from touching active or internal moving parts with tools, wires or similar foreign matter > than ø 1.0 mm.	4	against spray water 	Water splashing against the object from all directions must not have any harmful effect.
5	from deposit of dust 	Total protection of persons from touching voltage-carrying or internal moving parts. Protection of the object from harmful deposit of dust. Access of dust is not completely prevented, but dust is prevented from access in a quantity impairing the functioning.	5	against jet water 	A jet of water nozzled against the object from all directions must not have any harmful effect.
6	from access of dust 	Total protection of persons from touching voltage-carrying or internal moving parts. Protection of the object from access of dust.	6	against flooding 	Water of temporary flooding, as by heavy seas, must not enter the object in any harmful quantity.
			7	in dipped state 	If the object is dipped into water (0.15-1m) under the defined conditions of pressure and time, water must not enter it in any harmful quantity.
			8	in submerged state 	If the object is submerged in water under defined extremely conditions, water must not enter in any harmfull quantity.

Figure 6.2: IP Protection Classes (DIN 40050/IEC 529 Standard)

(ii) The introduction of 1 MB memory size as a subcategory in complexity driver d_4 (memory size). The 1 MB memory size introduced in the complexity driver d_4 (memory size) with an adjustment factor of 1.

(iii) The change in adjustment factors on the subcategories of complexity driver d_8 as follows: distributed functionality of 70% or more should have an adjustment factor of 1, distributed functionality between 40% and 70% an adjustment factor of 0.6 and distributed functionality of less than 40% an adjustment factor of 0.3.

Receiving the experts' feedback as it is discussed above, the researcher adopted the medium value of the three weightings for each complexity driver and he also introduced the changes requested on the subcategories and their adjustment factors as requested. After the changes were introduced, the final form of the weighting matrix is the one shown bellow:

Table 6.4: Amended weighted HW D&D Complexity coefficients

Driver	Name	Weight (%)	Value	New Factor
d_1	*Type of Components*	*20*	1-10 ICs No micros	0,4
			1-20 ICs No micros	0,6
			1-20 ICs 1 micro	0,8
			More than 20 ICs 2 micros	1
d_2	*Number of Components*	*10*	0 - 20	0,2
			21 - 50	0,4
			51 - 100	0,6
			101 - 300	0,8
			More than 300	1
d_3	*Memory type*	*5*	ROM	0,2
			RAM	0,3
			PROM	0,4
			EPROM	0,5
			OTP	0,8
			FLASH	1
d_4	*Memory Size*	*8.3*	8kb	0,2
			16kb	0,3
			32kb	0,4
			64kb	0,5
			128kb	0,6
			256kb	0,7
			512kb	0,8
			1MB	1
d_5	*Number of Interfaces*	*10*	0 to 10	0,2
			10 to 20	0,5
			20 to 30	0,8
			More than 30	1
d_6	*Type of Interfaces*	*18.3*	Digital Slow	0,2
			Ground Connections	0,2
			Digital Quick	0,3
			Communication Bus (LAN, CAN, etc)	0,4
			Differential Inputs	0,6
			Combination of the above	1
d_7	*Functionality Class*	*10*	A (e.g. Radio)	0,3
			B (e.g. Immobiliser)	0,7
			C (e.g. ABS)	1
d_8	*Distributed Functionality*	*11.7*	Less than 40% distributed	0,3
			40% to 70% distributed	0,6
			70% distributed	1
d_9	*Test/Acceptance Criteria*	*6.7*	IP Protection class (from 0 to 2)	0.2
			IP Protection class (between 2 and 4)	0.3
			IP Protection class (more than 4)	0.4
			Mechanical Requirements	0.5
			EMC Requirements	0.5
			Temperature Requirements/ Normal Temp (85%)	0.6
			Temperature Requirements/ Extreme Temp (85%)	0.7
			Water Resistance	0.8
			Combination of the above	1

The amended complexity metric, before it is presented here, was sent to the experts to examine it for its accuracy and therefore validate them. All experts were happy with the amendments introduced, therefore the complexity metric is onsidered validated.

6.3. Case studies

Having derived the complexity factors, their weighting, the complexity subcategories and their corresponding adjustment factors, the next step would be to apply this complexity factors to a number of case studies. By doing that, a complexity value would be calculated and plotted against corresponding HW D&D cost, in order for the researcher to be able to establish if there is any relation between HW D&D complexity and HW D&D cost.

Six case studies, all coming from the body domain of the car, were obtained from the sponsoring organisation's cost estimating department. These six case studies were selected because they both included a proprietary BOM and HW D&D cost as well, so they could be used both for deriving the HW D&D complexity value and for plotting this complexity value against the corresponding HW D&D cost.

For each of the 6 case studies, a BOM and its corresponding HW D&D cost were available. An example (the BOM for one of the 6 case studies) is presented in table 5.8. The name of the actual item is not included for confidentiality reasons.

Table 6.5: An example of a BOM (Case Study 1)

Component Type	Number of Components	Component Type	Number of Components
μC	1	symbol LED - blue	1
IC	1	symbol LED - green	16
IC	1	symbol LED - red	11
SMD connector	1	symbol LED - amber	7
Radial connector	1	diode green	12
capacitor	5	Diode	5
Connector	1	Other SMD	174
Varistor	1	Steppermotor	4
Cristal	1	TLE	1
Buzzer	1	PCB board	1
LCD 2x 14 character	1		
		Total No. of Components	**246**
		HW D&D cost	**0.356**

In order to derive the complexity metric, values for each one of the complexity drivers of table 6.4 should be obtained. The values for complexity drivers d_1 to d_4 were obtained by the researcher by manually counting the corresponding values from the above BOM. Values for complexity drivers d_5 to d_9 were obtained by the experts based on their experience, whereas the values for each item's HW D&D cost were obtained by the sponsoring organisation's engineering department, as these details are not displayed on the actual BOM. Summarising this information, the following table is derived:

Table 6.6: Complexity factor values for case study 1

Name	Weight	Value	Count	Factor
Type of Components	20.0	1-20 ICs 1 micro	2 ICs, 1 micro	**0.8**
No. Components	10.0	101 – 300	246	**0.8**
Memory type	5.0	FLASH	FLASH	**1**
Memory Size	8.3	256kb	256kb	**0.7**
No. of Interfaces	10.0	20 to 30	26	**0.8**
Type of Interfaces	18.3	Combination of the above	ALL	**1**
Functionality Class	10.0	B (e.g. Immobiliser)	B	**0.7**
Distributed Functionality	11.7	Less than 40% distributed	<40%	**0.3**
Test/Acceptance Crit.	6.7	Combination of the above	ALL	**1**

In order to obtain the complexity number for the above item, equation 1 is implemented:

Complexity number for item 1 (C1) =
$20*0.8+10*0.8+5*1+8.3*0.7+10*0.8+18.3*1+10*0.7+11.7*0.3+6.7*1$

which gives a complexity number of 78.32 for case study 1 (C1 = 78.32).

The same procedure is also applied for the remaining 5 items. The results are summarised in the following table:

Table 6.7: Complexity factor values for the 6 case studies

Driver	Weight	value for item 1	factor	value for item 2	factor	value for item 3	factor	value for item 4	factor	value for item 5	factor	value for item 6	factor
Type of Components	20.0	2 ICs, 1 micro	0.8	4 ICs, no micro	0.40	2 Ics, 1 micro	0.80	1 IC, 1 micro	0.80	4 ICs, 2 micros	1.00	5 Ics, 1 micro	0.80
No. Components	10.0	246	0.8	93.00	0.60	195.00	0.80	231.00	0.80	360.00	1.00	200.00	0.80
Memory type	5.0	FLASH	1	----	0.00	----	0.00	ROM	0.20	ROM	0.20	FLASH	1.00
Memory Size	8.3	256kb	0.7	-----	0.00	----	0.00	128kb	0.60	128kb	0.60	1MB	1.00
No. of Interfaces	10.0	26	0.8	9.00	0.20	5.00	0.20	8.00	0.20	35.00	1.00	6.00	0.20
Type of Interfaces	18.3	ALL	1	ALL	1.00	ALL	1.00	ALL	1.00	Digital Slow	0.20	ALL	1.00
Functionality Class	10.0	B	0.7	A	0.30	C	1.00	A	0.30	B	0.70	C	1.00
Distributed Functionality	11.7	<40%	0.3	<40%	0.30	<40%	0.30	>70%	1.00	<40%	0.30	<40%	0.30
Test/Acceptance Crit.	6.7	ALL	1	ALL	1.00	ALL	1.00	ALL	1.00	ALL	1.00	ALL	1.00
Complexity	(Factor *	Weight)	**78.32**		**47.51**		**64.51**		**74.68**		**66.85**		**77.81**
HW D&D Cost			**0.356**		**0.060**		**0.218**		**0.300**		**0.221**		**0.213**
Classification			**1**		**6**		**5**		**3**		**4**		**2**

In the following table, the complexity values are presented in descending order alongside with their corresponding D&D cost:

Table 6.8: Classification of case studies in descending order according to their complexity value

		Complexity	HW D&D cost (in Euros/piece)
1	item 1	78.32	0.356
2	item 6	77.81	0.213
3	item 4	74.68	0.300
4	item 5	66.85	0.221
5	item 3	64.51	0.218
6	item 2	47.51	0.060

Using the values of complexity and HW D&D cost as they are displayed in table 6.8 above, the following graph is derived:

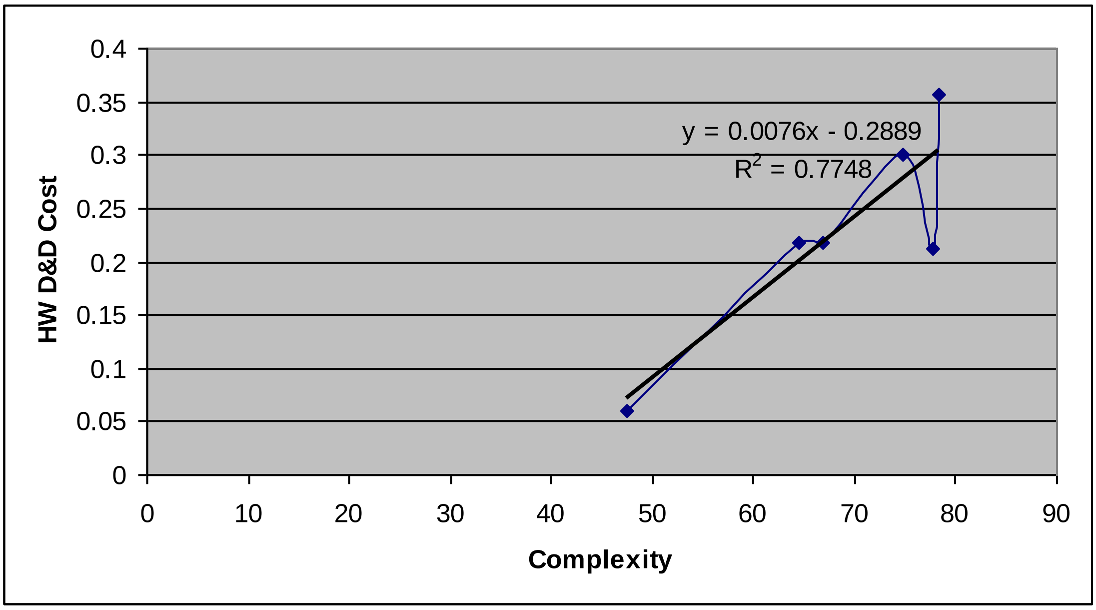


Figure 6.3: Complexity versus HW D&D cost.

6.3.1. Presentation of the results

The correlation graph presented in Figure 6.3 above has been derived using the Microsoft Excel spreadsheet. The correlation line shows that there is good correlation between the HW D&D complexity metric and the corresponding cost. This is also verified by the correlation coefficient R^2 value (R^2 = 0.7748) of the correlation equation (y=0,0076x-0.2889), since correlation coefficient values between 0,7 and 1 show that a fair amount of correlation exist (Croft et al, 1995; Bird, 2003). Therefore, the above correlation equation (y=0,0076x-0.2889) could be used to estimate the HW D&D cost of an embedded system, based on the HW's D&D complexity, as this has been defined in the earlier paragraphs of this chapter.

6.4. Framework Validation

The researcher, in order to validate the obtained results, performed 2 interviews with 2 experts from within the sponsoring organisation. Details of the 2 interviewees are presented bellow:

Table 6.9: Interview participants' details

	Job Title	**Years of Experience**
Expert 1	**System Engineer,** Body and Security Electronics, Electronic Subsystems, EESE	8 years
Expert 2	**Electronics Cost Estimator,** Finance	5 years

The first expert was interviewed in the sponsoring organisation's premises, whereas the interview with the second expert was performed over the phone, because the 2nd expert is located in a foreign country and it was extremely difficult for a face to face interview to be arranged.

Interviews were performed using a semi-structured questionnaire (found in Appendix G). The researcher opted for a semi-structured questionnaire in order to allow the experts to either expand on their answers or offer any additional comment if they thought this would be necessary. In the case of the second expert, the questionnaire was emailed to him and after 5 days he was contacted by the researcher for the telephone interview.

Interviews were carried out using the following procedure: In a first step, the researcher presented to the expert an introduction to the scope, the aim and the objectives of the interview, in order to 'set the scene' and familiarise the expert with the procedure. In the second stage, interviews were initiated with the use of the semi-structured questionnaire. At the end of the workshop, there was an open discussion about the HW D&D metrics and model in general.

Therefore, the interviews' objectives were:

1. Validate the approach presented by the researcher for obtaining HW D&D cost based on the HW Complexity value.
2. Validate the results presented by the researcher
3. Identify – if possible- improvements to the suggested model.

Experts' answers and comments have been captured on tape (for expert 1) and on paper (for expert 2) for validation reasons and for any future reference. A summary of the experts' answers is presented in table 5.12 at next page:

Table 6.10: A summary of the experts' answers

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested framework?	Yes, the model follows a logical sequence of steps.	Yes, the model follows a sound structure.
Are the results logical?	Yes, they are. This is also shown by the correlation coefficient R^2.	Correlation coefficient R^2 shows that there is good correlation between the HW D&D complexity and HW D&D cost.
Could the suggested framework be used to predict HW D&D cost based on the HW complexity?	Yes, it could, for items belonging to the body domain of the car.	Yes, it could.
Is there anything not covered from the model?	The areas of Infotainment and Chassis are not covered by the model.	Infotainment and Chassis
How is this model different than the one you currently use?	There is currently no model used for HW D&D effort estimation within the organisation.	No model is currently used to predict HW D&D effort.
What are the potential issues in implementing such a methodology?	Data (cost, complexity factors) have to be collected and stored, in order to be used for further calibration of the model, and for its expansion into covering the Infotainment and Chassis areas.	More than 6 case studies are necessary for coefficients' calibration.

6.4.1. Presentation of the results

As it is observed from the experts' answers as they are presented in table 5.12 above, since there was no objection against the rationale of the model or against any of the factors or the weightings, then the metrics created, the model presented and the results obtained are considered validated. However, more samples (case studies) have to be collected for expanding the model in covering – apart from the body car domain- the Infotainment and Chassis areas, and also, for model calibration.

6.5. Summary

The purpose of this chapter was twofold: first, to provide an in-depth analysis and improve understanding of relative contribution of HW, SW, Reuse and Integration efforts to the overall ES D&D effort. Secondly, to present the reader with the first part of the ES D&D effort estimation framework: the HW D&D effort estimation module. The first part is addressed through a workshop help in the sponsoring organisation premises, whereas the second point was addressed through the development of a framework that links the HW implementation details with its D&D cost. The next chapter, chapter 6, presents the development of the next module of the ES D&D effort estimation framework, the SW D&D effort estimation module.

7. Developing of a Cost Estimating Framework for Embedded SW Design and Development

Automotive OEMs provide their suppliers with specifications either in UML Use Cases or in Statecharts. As it was concluded earlier in chapter 2 for Use Case based specs, Use Case Points (UCP) method is the only applicable method to estimate SW development effort estimation. However, UCP has never been applied in the embedded software domain (Anda et al, 2002; Kusumoto et al, 2004; Ribu, 2001). The researcher decided to go forward and apply the UCP method to estimate the D&D of embedded SW in order to check if UCP could be a reliable estimation method for the embedded SW domain. UCP would only be applied to specifications expressed in UML Use Case models. Since there is so far no model for estimating development effort from Statecharts specifications, the researcher decided to go forward by developing a new effort estimating framework to cover this domain as well.

The next paragraphs describe the process followed by the researcher on (i) evaluating the applicability of the UCP method on estimating the development effort embedded SW and (ii) on developing metrics for estimating embedded SW D&D effort based on Statecharts specifications. The model is based in the following 2 assumptions:

1. There is at least 1 SW project whose design and development effort is known.
2. The functionality expressed by either the UML Use Case diagram or the Statecharts diagram is attributable to SW (Ziebart, 1991; Gupta, 2003).

The basic idea of the model is that since the effort of at least 1 project is known, if a metric can be derived to indicate the difference between the SW projects, then the productivity can be derived as (Greves and Schreiber, 1996):

Productivity (Prod) = (output performed) / (unit of human effort) =
= (metric) / (effort)   (eq.1)

The productivity can be assumed as constant for similar type of ES SW development and as a result, effort required for a new project can be estimated using the above equation.

7.1. Use Case based effort estimation: Case Studies

The researcher acquired 7 case studies from within the sponsoring organisation, all of them coming from the Information and Entertainment (Infotainment) domain, and he applied the UCP method as described in literature review. Each of the case studies consist of a document which contains the overall use case diagram, which describes the ES indented functionality. For example, for case study 4 (an AM-FM radio) the Use Case diagram is the following (for confidentiality reasons, the use case descriptions are not disclosed):

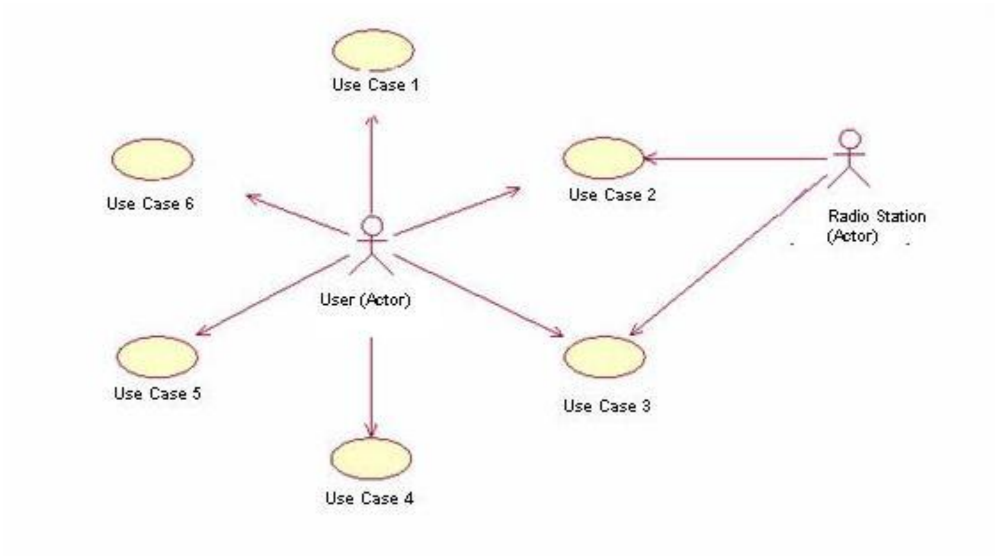


Figure 7.1: A Use Case diagram

In the same document, for each of the Use Cases that constitute the Use Case diagram there is a corresponding Use Case documentation. For example, the corresponding documentation for Use Case 1 is the following:

Purpose:
To define what shall happen when a user activates/deactivates the AM/FM application
Actors:
The User
Preconditions:
The Entertainment system is activ
Main Flow of Events

1. This UseCase starts when the user activates the AM/FM application via the HMI Input.
2. The current audio output shall be muted.
3. The AM/FM tuner shall recall the last saved parameters according to [GR-AM/FM tuner – parameters to be stored between driving cycles] and go to the RadioBand and the last tuned frequency. If the last tuned frequency was a preset the AM/FM tuner shall be in PresetMode or in AutoStoreMode, otherwise it shall be in NormalTuneMode.
4. **AM/FM TunerNormalView** shall be displayed.
5. Audio output shall be unmuted.
6. The user deactivates the AM/FM application via the HMI Input.
7. The AM/FM tuner shall store the current parameters according to [GR-AM/FM tuner – parameters to be stored between driving cycles].
8. Audio output shall be muted.
9. **AM/FM TunerNormalView** shall be removed.
10. The UseCase ends here

Figure 7.2: A Use Case documentation

The first step on the Use Case Points method is to identify the number and the type of the actors. Actors are classified as Simple, Average and Complex according to the type of its interaction with the system (please see paragraph ….). In the case of case study 4, there are 2 actors: the first one is the Radio Station and the second is the user. The radio station actor is classified as simple and therefore it is assigned a weight of 1, whereas the second actor, the user, is classified as complex (since human reactions are unpredictable) and therefore it is assigned the weight of 3. Therefore, the Unadjusted Actors Weight (UAW) is calculated as UAW = (1*1) + (1*3) = 4.

The second step on the Use Case Points method is to identify the number and type of the Use Cases. Use cases are classified as simple, average or complex according to the number of transactions (scenarios) the use case has. For example, in the case of case study 4, Use Case 1 has 10 transactions, which classifies it as complex. Therefore, Use Case 1 has is assigned a weight of 15. The same is done for all the 6 Use Cases of the Use Case diagram. By this, the following classification of Use Cases arises:

Table 7.1: Use Case classification for case study 4

Use Case	Category	Weight
1	Complex	15
2	Average	10
3	Complex	15
4	Complex	15
5	Complex	15
6	Complex	15

Therefore, the Unadjusted Use Cases Weight (UUCW) and the Unadjusted Use Case Points (UUCP) are calculated as follows:

UUCW = (1*10) + (5*15) = 85

and

UUCP = UAW + UUCW = 4 + 85 = 89

The next step in the Use Case Points estimation method is to estimate the Technology and Environmental Factors. Anda et al (2002), Kusumoto et al (2004) and Ribu (2001) when used the UCP method to estimate the effort for SW development, observed that (a) Technology Factors can be omitted, since they have already been taken into consideration when designing the requirements of the system, and (b) the estimation without the Technology Factors gave results very close to the expert's results. Based on the above, the researcher decided not to include the Technology Factors when applied the UCP method.

To estimate the Environmental Factor, values have to be assigned for each of the factors of Table 6.5. For case study 4, these values are the ones assigned on table 6.7 bellow. These values are set by the researcher on the best of his knowledge regarding the supplier's development environment. As it will be shown latter in this chapter, the experts validated these values.

Table 7.2: Environmental factors calculation for Case Study 4

	Factor	Weight (W)	Value (V)	W x V
1	Familiar with RUP	1.5	1	1.5
2	Application Experience	0.5	1	0.5
3	Object-Oriented Experience	1	1	1
4	Lead Analyst Capability	0.5	5	2.5
5	Motivation	1	5	5
6	Stable Requirements	2	5	10
7	Part-time workers	-1	0	-1
8	Difficult Programming Language	-1	2	-2
			Total:	**18.5**

Therefore, the Environmental Complexity Factor (EF) is calculated as:

EF = 1.4 + (-0.03*EF) = 1.4 + (-0.03*18.5) = 0.845

Finally, the Adjusted Use Case Points (AUCP) for case study 3 is:

AUCP = UUCP * ECF = 89 * 0.845 = 75

Since all 7 case studies come from the same supplier, this means that the development environment remains the same for all the 7 case studies and therefore the value for the Environmental Factor is the same for all of the 7 case studies. Based on the guidelines of the UCP method as they were presented in literature review, the final UCP values for the 7 case studies were the following (rounded to the closest integer):

Table 7.3: UCP count for the 7 case studies

Item	Actors	Weight	UAW	UC	Weight	UUCW	UUCP	EF	UCP
Case	Simple = 0			Simple = 4	4 x 5 = 20				
Study 1	Average = 0			Average = 9	9 x 10 = 90				
	Complex = 1	1 x 3 = 3	3	Complex = 10	10 x 15 = 150	260	263	0.845	222
Case	Simple = 0			Simple = 0					
Study 2	Average = 0			Average = 1	1 x 10 = 10				
	Complex = 1	1 x 3 = 3	3	Complex = 4	4 x 15 = 60	70	73	0.845	62
Case	Simple = 1	1 x 1 = 1		Simple = 2	2 x 5 = 10				
Study 3	Average = 0			Average = 1	1 x 10 = 10				
	Complex = 1	1 x 3 = 3	4	Complex = 1	1 x 15 = 15	35	39	0.845	33
Case	Simple = 1	1 x 1 = 1		Simple = 0					
Study 4	Average = 0			Average = 1	1 x 10 = 10				
	Complex = 1	1 x 3 = 3	4	Complex = 5	5 x 15 = 75	85	89	0.845	75
Case	Simple = 1	1 x 1 = 1		Simple = 2	2 x 5 = 10				
Study 5	Average = 0			Average = 9	9 x 10 = 90				
	Complex = 1	1 x 3 = 3	4	Complex = 5	5 x 15 = 75	175	179	0.845	151
Case	Simple = 0			Simple = 1	1 x 5 = 5				
Study 6	Average = 0			Average = 5	5 x 10 = 50				
	Complex = 1	1 x 3 = 3	3	Complex = 7	7 x 15 = 105	160	163	0.845	138
Case	Simple = 0			Simple = 0					
Study 7	Average = 0			Average = 5	5 x 10 = 50				
	Complex = 1	1 x 3 = 3	3	Complex = 13	13 x 15 = 195	245	248	0.845	210

For case study 1, the design and development effort was known and it was 1000 hours. Therefore, the Productivity can be calculated as:

Productivity (Prod) = (output performed) / (unit of human effort) =
= (metric) / (effort) = (222) / (1000) = 0.222 (UCP/hour)

Assuming the Productivity is constant, we can derive the design and development effort for the rest of the items:

Table 7.4: Estimation of D&D effort for UML Use Cases specifications

Item	UCP	Effort	Rounded Effort
Case Study 1	222	1000	1000
Case Study 2	62	279.2793	279
Case Study 3	33	148.6486	149
Case Study 4	75	337.8378	338
Case Study 5	151	680.1802	680
Case Study 6	138	621.6216	622
Case Study 7	210	945.9459	946

As it was discussed before, the major problems when counting UCP are (i) the way UCP have been documented and (ii) the productivity ratio. The first issue has been tackled by the sponsoring organisation by issuing a document with guidelines on how the Use Cases should be documented. This ensures consistency on Use Cases representation across the organisation. The second issue is tackled by the researcher by avoiding using any of the productivity ratios suggested in the literature; instead, the researcher derives a productivity ratio specific for the organisation, coming out from real-world data (sponsoring organisation's UC diagram and its known effort). The fact that all case studies come from the Information and Entertainment (Infotainment) domain, also ensures that productivity remains the same across all the SW projects, since as Putnam and Myers (2000) state productivity is different between different application areas.

7.2. Statecharts based metrics and effort estimation

A review in the literature and in the market reveals that there is no statechart based metric to link the SW's specifications with its development effort. In the development of the metric, apart from his own literature review and the feedback from experts when visiting companies, the researcher also involved experts from the automotive modeling domain. The researcher, alongside his literature review, conducted various Statecharts modeling experts from all over the world by email, asking them to participate in identifying the factors that affect a Statecharts diagram's complexity. At the end, 2 experts accepted to participate. The researcher conducted the 2 experts telephonically. Details of the 2 experts that participated in the development of the initial list of factors are presented below:

Table 7.5: Statechart metrics development experts

	Position	**Experience**
Expert 1	CAE and Model Based Systems Engineering, R and VT, Electrical and Electronics Systems Engineering	8 years
Expert 2	Senior Applications Engineer	5 years

Interviewing them consisted of two steps: In the beginning the experts were asked to name the Statecharts complexity factors, whereas in the second change they were asked to make any additional comment they thought necessary. Combining these sources of information (literature review and opinions of experts), the following list of factors was obtained:

1. Number of states
2. Number of events, variables and activities
3. Number of actions
4. Number of transitions
5. Number of levels in hierarchy
6. Number of parallel machines
7. Number of data types (integer, boolean, etc)
8. Number of mini-specs
9. Number of truth tables

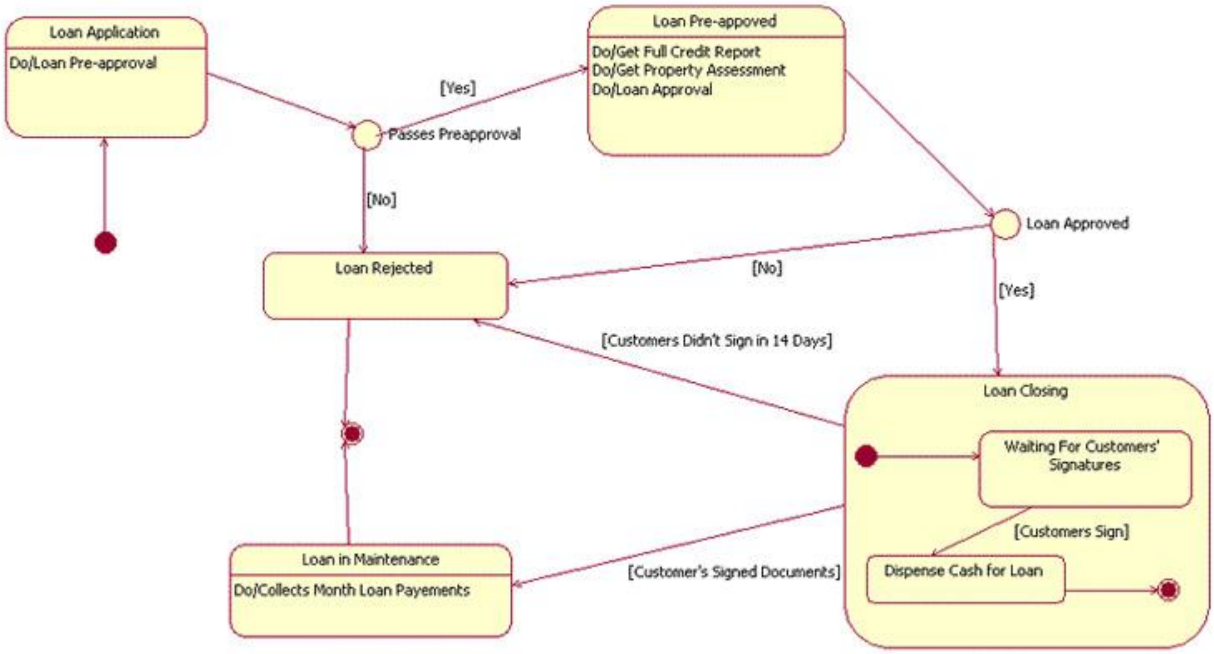


Figure 7.3: A statechart diagram (Bell, 2003)

A state is a condition or a situation an object possesses in a certain moment under certain circumstances (Rumbaugh et al, 2005). States are represented as rounded rectangles (figure 6.4). When an event arrives, the object leaves its current state and it proceeds into a next one. Transitions are shown as arrows (figure 6.4) and their complete syntax (each of the above parts is optional) is (Maciaszek, 2001; Fowler and Scott, 2002):

event(parameters)[guard]/action

To illustrate the context, consider the following example (Maciaszek, 2001):

mouse_button_clicked(right_button)[inside the window]/display second task bar

A guard is a condition which can be either 'right' or 'wrong', 'true' or 'false'. If guard is 'true', then the transition is carried out and the respective action is executed; if guard is 'false', then the transition is not carried out (or an alternative one is performed) (Fowler and Scott, 2002). An Activity is a computation located within a state and, in contrast with an action that is considered to take no time at all, it needs time to get executed (Fowler and Scott, 2002; Maciaszek, 2001).

When the decomposition in the system's hierarchy reaches the lowest level(s), then this lowest level(s) consists only of executable activities, written in actual form (code). This executable activity is then called a *mini-spec*. A *truth table* represents logical expressions (ie if A is true and B is true, then A and B is also true) and they are tied to activities, actions or subroutine procedures (Statemate Magnum manual). Parallel machines (or concurrent statecharts) are illustrated as in figure 7.4 bellow. Parallel machines are used when it has to be shown that an object has various independent behaviours (Fowler and Scott, 2002; Peters and Pedrycz, 1999); for example, in figure 7.4 bellow, the object could either be in state s11 or in state s12 before it proceeds to a next state.

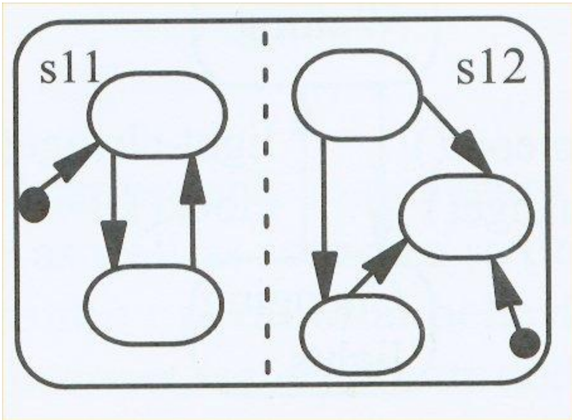


Figure 7.4: Parallel machines (Peters and Pedrycz, 1999)

The two engineers that contributed in the development of the above list, noted that the first 4 factors (number of: states, events and variables, actions, and transitions) are called *statistic* information, and their number can easily be counted by the statechart diagram. However, they stated that the complexity of a statechart diagram (and therefore of the system) is further affected by an additional set of factors; the number of: levels of the hierarchy, parallel machines, data types, mini-specs and truth tables. These are called *structural* information and their counting needs careful investigation of both the statechart diagram and the diagram's documentation as well.

Since structural information affect the complexity of the model, there should be weighting assigned to them in order to assess each structural factor's contribution to the system. To assess this, the researcher condacted both the sponsoring organisation as well as another automotive OEM, asking for experts to contribute on validating the list of factors and also offer weightings for the factors 5 to 9 (structural information).

Finally, 3 experts from the sponsoring organisation and 2 experts from the second automotive OEM participated. These experts were selected from the interviewed organisations because of their knowledge and expertiece in the areas of model based systems engineering and Electronics as the more appropriate people to offer feedback in this survey. A complete list with the participants' job roles, and years of experience can be found in the next table:

Table 7.6: Survey participants

	Position	Experience
Expert 1	CAE and Model Based Systems Engineering, R and VT, Electrical and Electronics Systems Engineering (EESE)	8 years
Expert 2	Senior Applications Engineer	5 years
Expert 3	Electrical CAE Research & Vehicle Technology	4 years
Expert 4	Manager, Systems Engineering	8 years
Expert 5	Project Leader, Systems Engineering	5 years

A questionnaire was developed by the researcher in order to serve as the medium for collecting the experts' answers. The questionnaire (which can be found at Appendix D) contains a set of questions that aimed to elicit the knowledge of the participating experts on if the suggested list of Statecharts complexity factors is complete or not and why and on assigning weights on the structural ones. There was also additional space on the questionnaire for the experts in order to express any additional information they think it would be necessary. To summarise, the survey's objectives were:

1. Validate the suggested list of Statecharts complexity factors, and
2. Assign weights on the structural complexity factors (number 5 to 9)

The questionnaire was e-mailed to the participants, who emailed back their answers to the researcher. When the information was collected they were analysed by examining the answers given in the questionnaire. The researcher looked for commonalities and differences between the experts' answers. An overview of experts' answers is being presented in the next table (Table 6.12):

Table 7.7: A summary of the participants' answers

Questions	**Expert 1**	**Expert 2**	**Expert 3**	**Expert 4**	**Expert 5**
Question 1	The list is complete.	The list is complete.	The list is complete.	The list is complete.	The list is complete.
Question 2	Number of levels in hierarchy: 35% Number of parallel machines: 30% Number of data types, Number of mini-specs and Number of truth tables: 35%	No weighting, it is up to the expert's knowledge to decide	Number of levels in hierarchy: 35% Number of parallel machines: 30% Number of data types, Number of mini-specs and Number of truth tables: 35%	Number of levels in hierarchy: 25% Number of parallel machines: 30% Number of data types, Number of mini-specs and Number of truth tables: 45%	No weighting, all factors are of equal importance (Therefore, each factor is assigned a weight of 33%)

Since all participating experts agreed that the suggested list of Statecharts complexity factors is complete, the list of factors is considered validated. As far as assigning weighting on the structural part of the list is concerned, based on the percentages given by the experts (Table 7.7) the researcher decided to adopt the mean value for each of the factors:

Number of levels in hierarchy = (35%+35%+25%+33%)/4=32%

Number of parallel machines = (30%+30%+30%+33%)/4=30,75%(rounded: 31%)

Number of data types
Number of mini-specs = (35%+35%+45%+33%)/4=37%
Number of truth tables

Combining the statistical factors and the weighed structural factors, the following metric can be derived:

Unadjusted Statecharts Complexity Metric (USCM) =

= (Number of states+ Number of events and variables+ Number of actions+
Number of transitions) (eq. 2)

and finally:

Adjusted Statecharts Complexity Metric (ASCM) =

= USCM x [(Number of levels in hierarchy) * 32%
+ (Number of parallel machines) * 31%
+ (Number of (data types + minispecs + truth tables) * 37%] (eq. 3)

7.2.1. Case Studies

The researcher acquired 3 case studies from within the sponsoring organisation, all of them coming from the Body domain, and he applied the method described in the previous paragraph. To ensure confidentiality, the Statecharts diagrams are not disclosed. The actual names of the items have also been changed.

Each of the case studies consists of a document in the form of figure 7.5 bellow. In the first level, the overall system's functionality is displayed in the form of a Statecharts diagram. Then, in the next levels, this system functionality is further decomposed using more detailed Statechart diagrams, until at the end, each statechart diagram represents the functionality of one and only activity.

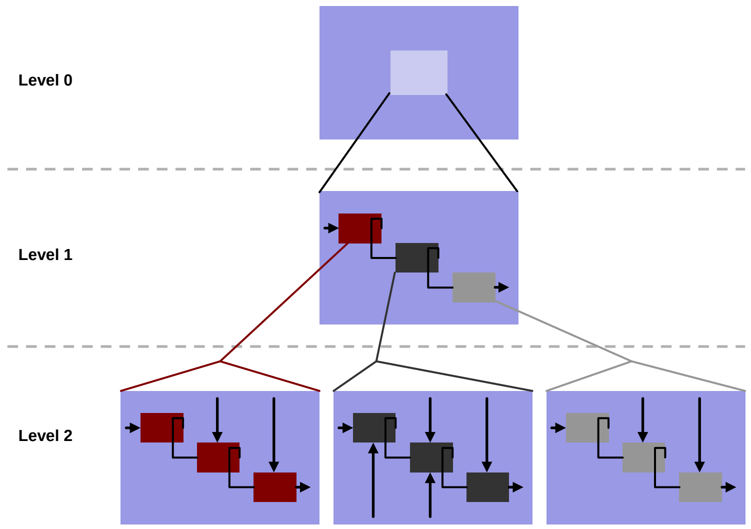


Figure 7.5: Statecharts specification structure (Rush, 2002)

Figure 7.6 presents the overall statecharts diagram for Case Study 1, whereas figure 7.7 presents the statechart diagram for activity 1.1.1 for the same case study:

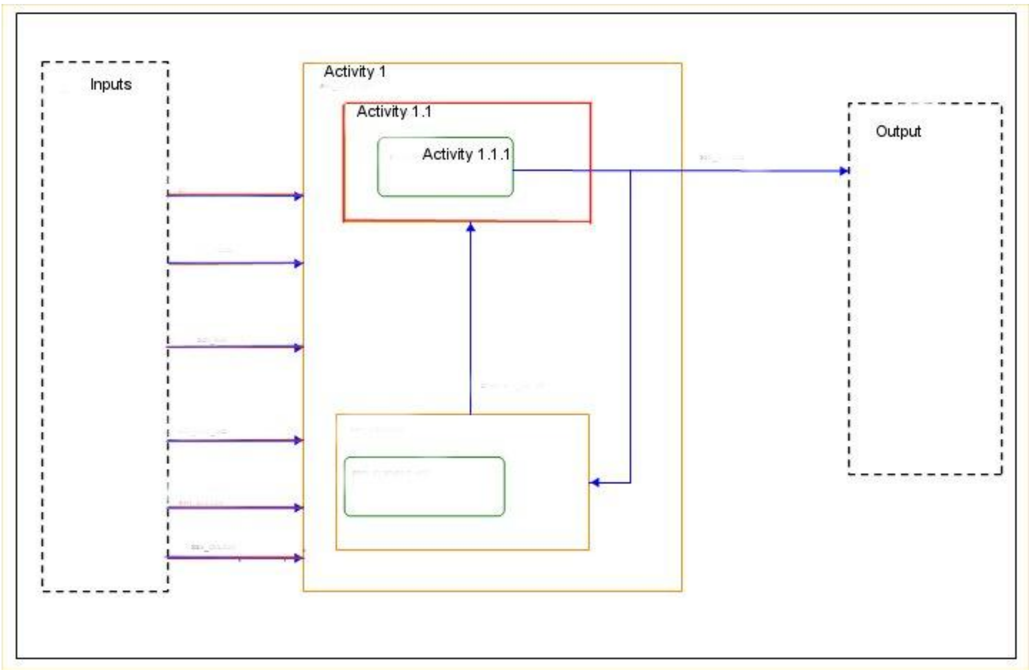


Figure 7.6: Overall Statechart diagram for Case Study 1

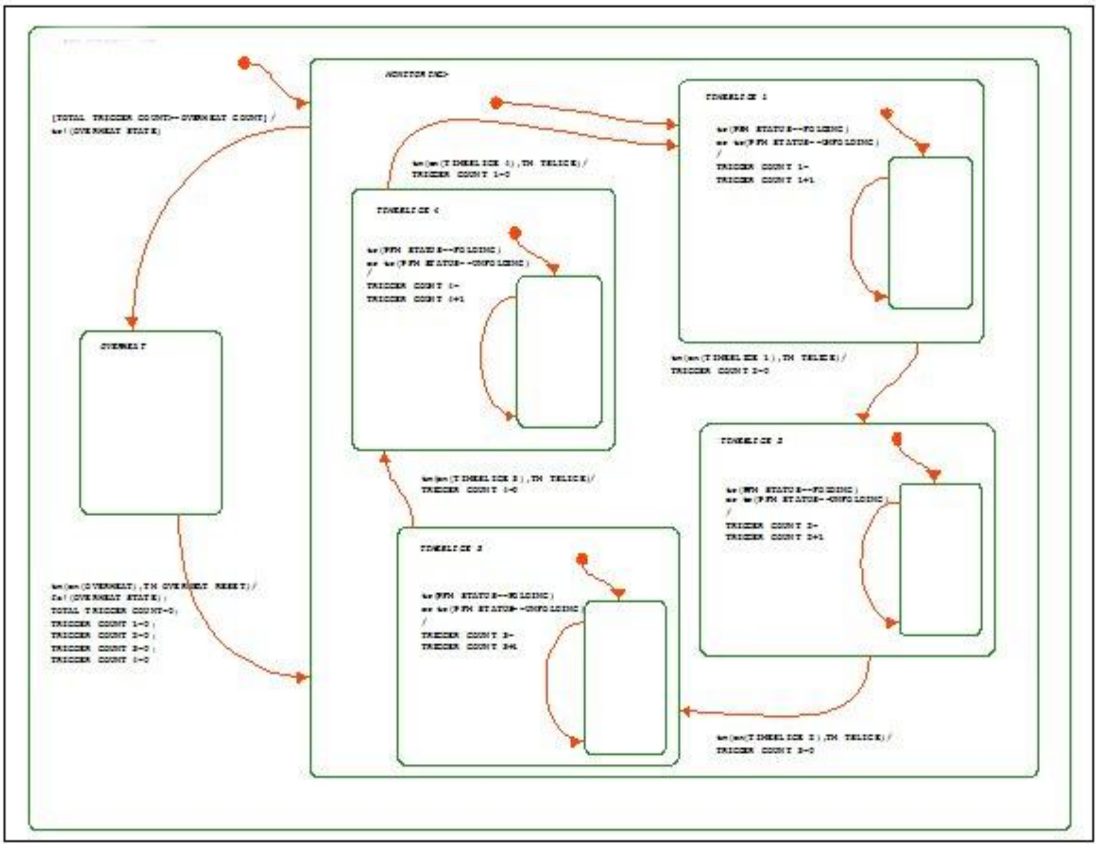


Figure 7.7: Statechart diagram for activity 1.1.1 for Case Study 1

In order for the Statechart Complexity number to be derived, values for each one of the statechart complexity factors have to be obtained. To obtain these values, the statechart diagrams of the lowest level are firstly examined. This is done because these diagrams contain on full detail all the necessary information needed (i.e. number of states, number of actions, etc). The values obtained for the statechart complexity factors for each of the lowest statechart diagrams are summed-up to derive the overall value for each of the complexity value for the complete system.

For Case Study 1, the researcher counted manually all the values for every statechart complexity factor, firstly for every lowest level complexity diagram and then, these values were summed up to derive the overall values for each factor for the complete system. Counting the values for each of the factors and applying equations 2 and 3 the following values were obtained for Case Study 1:

Table 7.8: Design and development effort estimation for Case Study 1

	Factors	Case Study 1
a	Number of States	13
b	Number of Events, Activities and Variables	16
c	Number of Actions	26
d	Number of Transitions	28
e	Number of Hierarchy Levels	4
f	Number of Parallel Machines	0
g	Number of Data Types	14
h	Number of Mini-Specs	0
i	Number of Truth-Tables	0
	USCM	83
	ASCM	536.18
	ASCM (Rounded)	536

The same procedure was also applied for the remaining 2 case study items. The results are summarised in the following table:

Table 7.9: Design and development effort estimation for all Case Studies

	Factors	Case Study 1	Case Study 2	Case Study 3
a	Number of States	13	82	68
b	Number of Events, Activities and Variables	16	163	130
c	Number of Actions	26	82	70
d	Number of Transitions	28	190	139
e	Number of Hierarchy Levels	4	4	3
f	Number of Parallel Machines	0	2	0
g	Number of Data Types	14	38	40
h	Number of Mini-Specs	0	0	0
i	Number of Truth-Tables	0	5	0
	USCM	83	517	407
	ASCM	536,18	9207,77	6414,32
	ASCM (Rounded)	536	9208	6414

Having derived the ASCM values for each of the case studies, the next step would be to use these values to estimate D&D effort for each one of them. To do this, the effort for D&D a SW project should be known. In this case, the effort for D&D item 2 was 800 hours. Using project's 2 AUCM value of 9208, the known effort of project 2 (800 hours) and equation 1, the rest of the D&D efforts were also estimated:

Table 7.10: Design and development effort estimation

Factors	Case Study 1	Case Study 2	Case Study 3
UUCM	83	517	407
AUCM	536,18	9207,77	6414,32
AUCM (Rounded)	536	9208	6414
Effort	46.56	800	557.25

7.3. Metrics and Results validation

The researcher, in order to validate the derived metrics and the obtained results, for both the Use Case Points and the Statecharts methods, organised a workshop at sponsoring organisation's premises. Two experts with considerable experience in both UML and Statecharts modelling participated, and their opinions were collected using a semi-structured questionnaire.

7.3.1. Workshop design

7.3.1.1. Target Audience

The aim of this workshop was to validate the derived metrics and the obtained results. In order to accomplish that, the researcher organised a workshop in the sponsoring organisation's premises with 2 experts in the modelling domain. It has to be stated here that their experience and therefore their opinions reflect the experience gained within the automotive sector only and from past projects accomplished within the sponsoring organisation.

These experts were selected from the sponsoring organisation since they were considered to be the people with the most experience and expertiece on modelling within the sponsoring organisation, and therefore they would provide an in-depth and detailed view of the situation. A list with the participants' job roles, and years of experience can be found in the following table:

Table 7.11: Survey participants

	Position	**Experience**
Expert 1	CAE and Model Based Systems Engineering, R and VT, Electrical and Electronics Systems Engineering (EESE)	8 years
Expert 2	Electrical CAE Research & Vehicle Technology	5 years

7.3.1.2. Questionnaire Development

A qualitative validation approach was followed, because of the lack of data within the OEM to support a quantitative one. A semi-structured questionnaire was developed by the researcher in order to serve as a guide on keeping the workshop on track. It contained a set of questions that aimed to assist experts on validating what was presented to them. It also served as a 'guide' of keeping the conversation 'in-track', to allow the participants to expand their views, and for the researcher, to make follow-up questions based on the received answers.

So, the workshop's objectives were:

1. Validate the metrics and the obtained results
2. Identify any improvements to the suggested model.

7.3.1.3. Conducting the Workshop

As it was described earlier, the researcher followed a semi-structured interview approach. This enabled the researcher to define the depth of the answers provided by the experts and for the experts to expand their answers. The workshop was performed on-site of the sponsoring organisation's location. All the information regarding the workshop has been captured in paper, to ensure accuracy on interpretation and analysis of results, as well as a point of reference when a doubt occurs.

The workshop was carried out following the procedure described bellow: initially, the researcher presented to the experts an introduction to the scope, the aim and the objectives of the workshop, in order to 'set the scene' and familiarise the experts with the procedure to be followed. After this initial stage, the workshop was initiated with the use of the semi-structured questionnaire. The researcher took the experts through one case study of UML Use Cases and one for Statecharts, explaining n detail the metrics development, the counting procedure and the results obtained and ask them to validate all these. At the end of the workshop, there was an open discussion about the modelling metrics in general. A summary of the experts' answers is presented in table 7.12:

Table 7.12: A summary of the workshop participants' answers

Questions	Participant 1	Participant 2
	Expert 1	Expert 2
Is the assumption that the functionality captured in a UML Use Case or in a Statechart diagram is mostly attributable to SW correct?	Yes, in a very big percentage	Yes, in its biggest part
Does Productivity change across different application areas and/or domains?	Yes, it changes in both cases. For example, Use Cases for Body electronics are much more detailed than for Infotainment	Yes, it changes in both cases.
Are the results realistic in terms of effort?	No, they are not, because there is a mistake on counting data types.	No, data types have been counted wrongly.
Is there anything not covered from the model?	No, the list of factors is complete	The list of factors is complete
Are the values assigned to factors realistic?	Data types on the statechart related metrics have been calculated wrongly. Only how many different types there are should be counted, not their amount	Wrong calculation of Data types. Only the number of their types should be counted.
How is this model different than the one you currently use?	There is currently no model used for SW D&D effort estimation within the organisation.	No model is currently used. SW D&D effort is confronted as a fixed percentage on top of the manufacturing cost.
What are the potential issues in implementing such a methodology?	It would considerably assist the estimation of SW D&D effort	It will provide a structured and easy to use way of predicting the SW D&D effort.

Correcting the mistake that was pointed out by the experts, the new values for data types, USCM, ASCM and efforts were re-estimated. The new values are shown bellow:

Table 7.13: Amended Design and development effort estimation

	Case Study 1	Case Study 2	Case Study 3
Number of Data Types	7	10	10
USCM	83	517	407
ASCM	321,21	3851,65	1896,62
ASCM (Rounded)	321	3852	1897
Efforts	66.66	800	393,97

The new values were shown to the experts, who were happy with the obtained results. In addition, the experts also validated the assumption that in the embedded systems domain the functionality represented by the Statecharts diagram is attributable to SW. Therefore, the framework is considered validated. However, the model has only been applied within the sponsoring organisation and its applicability and transferability within other OEMs or industries has to be examined in the future.

7.4. Summary

In this chapter, the development of the second part of the ES D&D effort estimation framework, the SW module, is presented. Since ES specifications come in two formats, Use Cases and Statecharts, separate modules have to be developed, in order for each specification case to be supported. In the case of Use Cases based specification, the Use Case Points method was implemented, whereas, for Statecharts based specifications metrics and procedures were developed. The two proposed approaches were validated by the experts. In the next chapter, chapter 7, the Reuse and Integration modules are presented.

8. Understanding Reuse and Integration

According to what has been presented so far (Literature and AS-IS models capture), reuse estimation constitutes a major problem when trying to estimate the effort for designing and developing an electronic item because the amount of reuse applied on the design and development of an electronic item (either in the SW or in the HW) can not easily be predicted. The reason for this is that (i) in the specification stage there is no detailed information on how the design is to be implemented and (ii) because this information (implementation details for both SW and HW) is protected by suppliers' IPR. Therefore, none of the above presented metrics (Table…) could not be applied within an automotive OEM to estimate the amount of SW and/or HW reuse, since the OEM has no access to the information required for their application. In addition, the metrics are limited to SW reuse only.

Today, the reuse issue is tackled by the OEM's experts by applying expert judgement. In the D&D effort capture workshop presented at the beginning of chapter 5, the engineers stated that in order to derive an estimation for the percentage of reuse contained in an ES, they first use their understanding to figure out how the system fulfils its functionality. Then, based on this understanding, they apply expert judgment in order to derive a SW and HW reuse percentage. This process however is performed in an ad-hoc manner, based each time on the experience of the expert that performs this procedure. Therefore, there should be a framework to make the reuse estimation procedure more transparent, structured and standardised. The development of such a framework is being described in the following paragraphs.

8.1. Developing a Reuse Estimation Framework

In order for the researcher to proceed on creating a reuse estimation framework, he first had to identify – based on his understanding - what are the cases reuse could be applied within an OEM and under what circumstances it occurs.

An embedded system could be produced for the OEM either by an existing supplier (a supplier already providing this specific product, even in different versions of the same carline) or by a totally new supplier. There is however also the case of having an existing supplier delivering a different product to the OEM than the one he was already supplying. In the case the supplier is totally new, the percentage of potential reuse would be expected to be very low, since the supplier would have no working experience with the OEM.

The opposite would be expected if the supplier is an existing one, since he will be holding cumulative experience with the OEM's standards, procedures, interfaces, etc, and therefore, the amount of reuse would be expected to be high. Finally, in the case that the supplier is an existing to the OEM supplier but for a different product, then the expected amount of reuse would be expected to be somewhere between the other two cases, since the supplier, although familiar with the OEM's practices and standards would still have to ''learn'' the specific requirements of the new project.

An embedded system could be used in an already existing platform (a specific car model on a specific carline), on a completely new platform, or in a different platform of the same carline. In the first case, the amount of reuse is expected to be high, since the system would ''sit'' in an already known environment. The opposite would happen in the second case. In this case, the amount of reuse is expected to be low, because the target environment is completely new. In the third case, the expected reuse percentage would sit somewhere in between the first and second case's, because although the carline would be the same, the platform would have different characteristics that the one the system was originally placed to.

Finally, an embedded system could be designed only for the OEM, or it could have also been designed for another automotive company as well. There is also the case that not all but some parts of the system to have been designed for another organisation. In the case of the system been designed only for the OEM, then the expected amount of reuse would be very low, since the system would be taylor-made only for the OEM. If the system has also been designed for another organisation, then the expected amount of reuse would be high. If the third case occurs, when some parts of the system have also been designed for another organisation, then the expected reuse percentage would be something between the reuse percentages of the other two cases. After identifying how and when reuse could occur within the OEM suppliers, the researcher summarised and categorised the 3 reuse cases as following:

1. Reuse percentage expected if the embedded system has been designed by a new or an existing supplier:

    - Low (Totally new supplier)

- Medium (An already existing to the sponsoring organisation supplier but for a different product)
- High (An already existing to the sponsoring organisation supplier, for the same product)

2. Reuse percentage expected if the embedded system has been designed for an existing platform or for a new platform:

   - Low (Totally new platform)
   - Medium (Different editions of the same platform)
   - High (Existing platform)

3. Reuse percentage expected if the embedded system has been designed only for the sponsoring organisation or it has also been designed for someone else:

   - Low (Designed only for the sponsoring organisation)
   - Medium (Some of its parts have been designed for other organisation(s) as well)
   - High (The biggest part of the system has been redesigned for other organisation(s) as well)

Table 8.1: Reuse Framework Matrix

Embedded System ''X''	Question 1: New/existing supplier?		Question 2: New/existing platform?		Question 3: Designed for someone else?	
	HW	SW	HW	SW	HW	SW
Low						
Medium						
High						

The underlying rationale of this approach is that tables like table 8.1 (''Reuse Table'') would be created for each and every one of the embedded systems the OEM procures from its suppliers. Each cell on table 8.1 represents a different situation regarding reuse, and each of these situations can only be recognisable and addressed by experts who are aware of the item under investigation. Therefore, reuse information regarding each of the potential circumstances described in table 8.1 could be collected and stored using table 7.2 for each and every system by the expert(s) that is responsible for each individual system. This way, there would always be a way of assessing reuse for every system and the whole reuse estimation process would become more structured and accurate. At the end, for the Reuse percentage estimation, for both HW and SW, the expert would choose the highest percentage, since this strategy gives the best business opportunity for the sponsoring organization.

Since only the experts from within the OEM could judge any particular item their organisation procures and since the sponsoring organisation's embedded systems' specifications can not be disclosed to any third party, for the validation of the presented here reuse estimation approach and for obtaining reuse percentages the researcher contacted engineers from within the sponsoring organisation.

8.1.1. Conducting the interviews

The researcher interviewed 2 experts from the sponsoring organisation. Interviews were performed with the use of a semi-structured questionnaire (found in Appendix H). Details of the 2 experts are given in table 7.3 bellow:

Table 8.2: Experts details

Name	Job Role	Experience
Engineer 1	**System Engineer,** Body and Security Electronics, Electronic Subsystems, EESE	8 years
Engineer 2	**System Engineer,** Safety Electronics R&VT, Electronic Subsystems, EESE	5 years

Both experts come from the Electronic Subsystems department. In order for the Reuse module to be developed, case studies should be used. So far in this thesis 16 case studies have been used, 6 for the development of the HW D&D effort estimation module (chapter 6) and 10 for the development of the SW D&D effort estimation module (chapter 7) of the ES D&D effort estimation framework. Due to the limited availability of the experts' time, it was not possible for all the 16 case studies to be used in the Reuse module development; only 3 out of them were used instead. These 3 case studies are the 3 case studies that were used for the development of the Embedded SW D&D effort estimation module based on statecharts specifications (for the case study details, please refer to paragraph 7.3.1)

The experts were sent by email a semi-structured questionnaire (found in Appendix H) and specifications of the 3 case studies which reuse percentages had to be obtained for. After a limited period of time the researcher contacted telephonically the experts and interviewed them. Interviewing the experts consisted of 2 steps: In the beginning, the expert was introduced to the framework's rationale and he was reminded of the specifications of the 3 case studies. In a second stage the expert was asked to supply reuse percentages for each of the 3 case studies. This way of interviewing was chosen because the Electronic Subsystems department (and therefore the two experts) resides in a country other than the country this research is performed, and meeting them face to face was not possible. The experts' comments were captured on paper for future reference and for validation purposes.

A summary of the experts' answers is presented in table 8.3 whereas in table 8.4 the reuse percentages both experts provided are displayed. There was a comment by both experts on question ''Is it going to be designed by a new or an existing supplier'', which was included in the reuse matrix on section 2 of the reuse estimation framework validation questionnaire (Appendix H). Both experts commented that the answer to this question can not be 'translated' to reuse levels' (therefore 'reuse percentages' can not be predicted), since the answer to this question only does not supply any important information for the percentages to be derived.

However, the answer to this question reveals the level of the cost impact on the embedded system's price if this was designed by a supplier who supplies the same organization with the same product. According to the first expert, if the system is going to be designed by a totally new supplier, then there would be a cost impact (to the price the supplier of the same product would ask) of 90%, since the supplier would have to familiarise himself with the standards, the procedures, the customisations of the organisation's electronic structure.

If the supplier is an existing to the sponsoring organisation supplier but for a different product, then the cost impact would be of a 50%, whereas if the supplier is an already existing to the sponsoring organisation supplier for the same product, then the cost impact would be of 10%, since, even though the supplier supplies the same product, there would always be the need for calibrating some parameters or for some testing.

The remaining expert noted that he was not able to provide exact percentages, however he agreed with the first expert's opinion that there is a different cost impact to the system's price depending on how familiar a supplier is with the procedures, the standards and the customisations an individual organisation requires. Under these observations, question ''Is it going to be designed by a new or an existing supplier'' was removed and the experts provided reuse percentages only for the two remaining questions (Appendix H).

Table 8.3: Summary of the experts' answers

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested framework?	Yes, the framework follows a logical approach.	It is a sound framework and offers a standardised estimation approach.
Could you please give your reuse percentages for each one of the 3 case studies?	These percentages are presented in table …	These percentages are presented in table …
Could the suggested framework be used to predict Reuse?	Yes, it can. It consists a more structured way of estimating reuse.	Yes, since we have no access to any other information.
Is there anything not covered from the model?	No, the framework is complete.	Since detailed information (LOC, implementation details, etc) is not accessible, this is the only way to currently access the reuse issue.
How is this model different than the one you currently use?	There is currently no model used for reuse estimation within our organisation.	No model is currently used to predict reuse; reuse is estimated based on expert judgement.
What are the potential issues in implementing such a methodology?	Data (reuse percentages from experts) have to be collected and stored.	Engineers have to give percentages for each and every item
Any additional comment?	It is not possible to estimate reuse (either for HW or SW) only by the fact that a system would be designed either by a new or by an existing supplier (Question 1 of the 'reuse matrix' of questionnaire H)	If the item is designed by a new or an existing supplier has a cost impact on the organisation, which is particular to the organisation itself. However, it is not possible for a reuse percentage to be estimated based only on that fact (Question 1 of the 'reuse matrix' of questionnaire H).

Table 8.4: reuse percentages provided by the experts

	Question 1: Is it going to be designed for a new or an existing platform?				Question 2: Has it already been designed for someone else?			
Case Study 1	**Expert 1**		**Expert2**		**Expert 1**		**Expert2**	
	HW	**SW**	**HW**	**SW**	**HW**	**SW**	**HW**	**SW**
Low	10	10	10	10	80	90	80	90
Medium	20	20	20	20	80	90	80	90
High	30	30	80	90	80	90	80	90
Case Study 2	**Expert 1**		**Expert2**		**Expert 1**		**Expert2**	
	HW	**SW**	**HW**	**SW**	**HW**	**SW**	**HW**	**SW**
Low	60	60	60	60	90	75	90	60
Medium	75	75	75	75	90	75	90	60
High	75	75	90	75	90	75	90	75
Case Study 3	**Expert 1**		**Expert2**		**Expert 1**		**Expert2**	
	HW	**SW**	**HW**	**SW**	**HW**	**SW**	**HW**	**SW**
Low	20	20	20	20	95	50	95	50
Medium	30	30	30	30	95	50	95	50
High	50	50	95	50	95	50	95	50

8.1.2. Results Presentation

As it can be derived by the experts' answers (Table 8.3) the presented in this chapter 'Reuse Framework' provides visibility on the reuse issue and offers a structured and standardised way for predicting reuse percentages for embedded systems used within the sponsoring organisation.

It is also observed that the reuse percentages expert 2 provided are almost the same with the percentages provided by expert 1. However, two differences are observed: the first difference is that the percentages expert 2 assigned to the ''High'' category for the 1st question are the same with the percentages assigned as ''High'' in the 2nd question. When expert 2 was asked during the interview why he assigned his reuse percentages in this way, expert 2 answered that since the item has already been designed for other organisation(s) as well (this means ''High'' in question 2) it would have been already designed for the existing platform of the sponsoring organisation (which is ''High'' in question 1). The second difference is on the ''Low'' and ''Medium'' percentages assigned in the second question for SW reuse in case study 2.

Regarding the first difference, the researcher decided to adopt the approach suggested by the second expert, since it describes a logical situation. In case of the second difference, the researcher decided to adopt the average between the two values. Following these guidelines, the following ''Reuse Table'' for the 3 case studies is created:

Table 8.5: Reuse matrix for the 3 case studies

		New/existing platform?		Designed for someone else?	
		HW	SW	HW	SW
Case Study 1	Low	10	10	80	90
	Medium	20	20	80	90
	High	80	90	80	90
Case Study 2	Low	60	60	90	60
	Medium	75	75	90	67.5
	High	90	75	90	75
Case Study 3	Low	20	20	95	50
	Medium	30	30	95	50
	High	95	50	95	50

8.1.3. Results Validation

The researcher, in order to validate the obtained results, performed 2 interviews with 2 experts from within the sponsoring organisation. Details of the 2 interviewees are presented bellow:

Table 8.6: Survey participants' details

	Job Title	**Years of Experience**
Expert 1	**System Engineer,** Body and Security Electronics, Electronic Subsystems, EESE	5 years
Expert 2	**Electronics Cost Estimator,** Finance	5 years

The first expert was interviewed in the sponsoring organisation's premises, whereas the interview with the second expert was performed over the phone, because the 2nd expert is located in a foreign country and it was extremely difficult for a face to face interview to be arranged.

Interviews were performed using a semi-structured questionnaire (found in Appendix J). The researcher opted for a semi-structured questionnaire in order to allow the experts to either expand on their answers or offer any additional comment if they thought this would be necessary. In the case of the second expert, the questionnaire was emailed to him and after a limited period of time the expert conducted him telephonically.

Interviews were carried out using the following procedure: In a first step, the researcher presented to the expert an introduction to the scope, the aim and the objectives of the interview, in order to 'set the scene' and familiarise the expert with the procedure. In the second stage, the researched described the case studies to the experts, in order to make them familiar with the commodities used for the Reuse module development. At the end, in the third stage, interviews were initiated with the use of the semi-structured questionnaire. At the end of the workshop, there was an open discussion about the ''Reuse Tables'' in general.

Therefore, the interviews' objectives were:

4. Validate the approach presented by the researcher for creating Reuse Tables.
5. Identify – if possible- improvements to the suggested model.

Experts' answers and comments have been captured on tape (for expert 1) and on paper (for expert 2) for validation reasons and for any future reference. A summary of the experts' answers is presented bellow:

Table 8.7: Summary of the experts' answers

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested framework?	Yes, the framework follows a logical sequence of steps.	Yes, the framework has a sound structure.
Are the results logical?	Yes, they are.	The results seem logical.
Could the suggested framework be used to predict Reuse (for both HW and SW)?	Yes, it could.	The suggested approach could support the estimation of HW and SW reuse.
Is there anything not covered from the model?	No, the framework is complete	The framework seems complete.
How is this model different than the one you currently use?	There is currently no model used for reuse estimation within the organisation.	No model is currently used to predict Reuse.
What are the potential issues in implementing such a methodology?	Data (reuse percentages) have to be collected. Reuse tables could also be created for each individual supplier.	Data (reuse percentages) have to be collected by the engineers.

Since there was no objection raised against the Reuse framework developed and the approach suggested, therefore, the Reuse framework is considered validated. However, for its full deployement, data (reuse percentages) have to be collected and introduced into the corresponding Reuse tables.

8.2. Integration

8.2.1. Introduction

Integration is a major issue within automotive industry; in fact, integration is the main reason for the continuously growing embedded systems D&D costs (Bouyssounouse and Sifakis, 2005). This is due to the fact that it is extremely difficult for OEMs to manage the integration of various subsystems towards the overall car electronics architecture, because these subsystems come from different suppliers with different design methods, different architectures (for both HW and SW), different (often proprietary) Real Time Operating System (RTOS), etc (Sangiovanni-Vincentelli, 2005).

In the embedded systems D&D effort capture workshop, presented in the beginning of chapter 5, experts stated that Integration effort cannot be predicted, as it is a case-by case issue. As they stated, there is no way of knowing before hand if the embedded system's SW might cause a problem in HW – or vice versa – or if the embedded system will work accordingly when integrated in the car's electronic architecture. All the above require additional effort in order to be corrected. However, because this effort depends entirely on the individual embedded system examined, engineers confront Integration effort using a fixed percentage on top of the HW and SW D&D effort, which for the sponsoring organisation's experts is 55%, as shown in the embedded systems D&D effort capture workshop.

Because it is very difficult to quantify the numbers, the researcher using expert judgement and through literature review performed a break-down on the Integration issue, in order to find out the factors that affect the Integration issue, offer more visibility and improve its understanding. This approach is presented in the following paragraphs.

8.2.2. Developing an approach for improving Itegration's Visibility.

From what has been presented so far, integration effort presents a great difficulty on its estimation, because in each of the 3 cases (HW to SW, SW to SW and Into the car integration) it depends on factors that can not be estimated before hand. This was also reflected on the experts' opinions: in the D&D effort capture workshop the experts stated that because integration is a case by case issue, they estimate its effort as an on-top to HW and SW D&D effort percentage (55%).

Because it is very difficult to quantify the numbers, the researcher, through his own understanding, created a break-down table for integration (table8.9). In a second step, in order to obtain weights for each of the integration categories and subcategories, the researcher conducted three experts from within the sponsoring organisation. Details of the 3 interviewees are presented bellow:

Table 8.8: Details of experts

	Job Title	**Years of Experience**
Expert 1	**System Engineer,** Security & Convenience, Electronic Subsystems, EESE	5 years
Expert 2	**System Engineer,** Safety Electronics R&VT, Electronic Subsystems, EESE	5 years
Expert 3	**System Engineer,** V34X Chassis Electronics, Electronic Subsystems, EESE	7 years

All experts were interviewed telephonically, because they were located in a foreign country and it was extremely difficult for a face-to-face interview to be arranged. Interviews were performed using a semi-structured questionnaire (found in Appendix I). The researcher opted for a semi-structured questionnaire in order to allow the experts to either expand on their answers or offer any additional comment if they thought this would be necessary. The questionnaire was emailed to the experts and after a limited period of time the experts were conducted by the expert telephonically in order for them to be interviewed.

Interviews were carried out using the following procedure: In a first step, the researcher presented to the experts an introduction to the scope, the aim and the objectives of the interview, in order to 'set the scene' and familiarise the experts with the procedure. In the second stage, interviews were initiated with the use of the semi-structured questionnaire. At the end of the interviews there was an open discussion about the Integration issue in general.

Experts' answers and comments have been captured on paper, for validation reasons and for any future reference. A summary of the experts' answers is presented on table 8.10.

Table 8.9: Integration brake-down matrix

	Integration Brake – Down Table		
Category	**SW to SW**	**SW to HW**	**In the car**
Category's percentage			
Subcategories and their percentage	application SW with: - RTOS:	''complete'' SW to underlying HW:	complete embedded system into the car's electronic architecture:
	- Drivers:	- Interface Development:	- Interconnection of the embedded system with other systems developed by different suppliers:
	- FNOS:		
	- Libraries:		
	- Other:		

Table 8.10: A summary of the experts' answers

Questions	Expert 1	Expert 2	Expert 3
Do you agree with the Integration categories and sub-categories as presented in the Integration breakdown matrix?	Yes, I agree. Both categories and subcategories are complete.	Yes, the presented Integration breakdown is complete and offers more visibility on the whole issue.	Yes, the presentation of categories and sub-categories is complete
Could you please give your percentages for each of the Integration 3 cases and for their sub-categories?	SW to SW: 30% SW to HW: 20% In the car: 50% I cannot assess sub-categories. This depends on each individual combination at the time.	SW to SW: 30% SW to HW: 20% In the car: 50% Sub-categories is a case by case issue, and therefore percentages can not be set.	SW to SW: 25% SW to HW: 25% In the car: 50% Sub-categories is a case by case issue, no generic percentage can be set.
Any additional comment?	No.	The integration sub-categories percentages cannot be predicted in a general manner, since it is a case by case issue.	Only the three main categories can be assigned a generic percentage; subcategories depend on each case.

8.2.2.1. Presentation of the results

As it is observed from the experts' answers as they are presented in table 8.10, since there was no objection against the suggested Integration brake down and the suggested Integration assessment approach, then, the approach presented and the results obtained are considered validated. As generic percentages for the 3 integration categories, the researcher decided to adopt the following percentages, because: (i) they are the most observed percentages between the 3 experts, and (ii) there is no big difference in the percentages given between the first two and the third expert (please see table 8.10).

Table 8.11: Survey participants' percentages

SW – SW	SW – HW	In the car
30%	20%	50%

8.2.3. Results validation

The researcher, in order to validate the obtained results, performed 2 interviews with 2 experts from within the sponsoring organisation. Details of the 2 interviewees are presented bellow:

Table 8.12: Survey participants' details

	Job Title	Years of Experience
Expert 1	**System Engineer,** Security & Convenience, Electronic Subsystems, EESE	5 years
Expert 3	**System Engineer,** V34X Chassis Electronics, Electronic Subsystems, EESE	7 years

The first expert was interviewed in the sponsoring organisation's premises, whereas the interview with the second expert was performed over the phone, because the 2nd expert is located in a foreign country and it was extremely difficult for a face-to-face interview to be arranged.

Interviews were performed using a semi-structured questionnaire (found in Appendix K). The researcher opted for a semi-structured questionnaire in order to allow the experts to either expand on their answers or offer any additional comment if they thought this would be necessary. In the case of the second expert, the questionnaire was emailed to him and after a limited period of time the expert conducted him telephonically.

Interviews were carried out using the following procedure: In a first step, the researcher presented to the expert an introduction to the scope, the aim and the objectives of the interview, in order to 'set the scene' and familiarise the expert with the procedure. In the second stage, interviews were initiated with the use of the semi-structured questionnaire. At the end of the interviews, there was an open discussion about the Integration issue in general. Therefore, the interviews' objectives were:

1. Validate the presented by the researcher methodology.
2. Validate the results presented by the researcher.
3. Identify – if possible- improvements to the suggested model.

Experts' answers and comments have been captured on tape (for expert 1) and on paper (for expert 2) for validation reasons and for any future reference. A summary of the experts' answers is presented bellow:

Table 8.13: A summary of the expert's answers

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested approach?	The suggested approach is a logical approach on understanding the integration issue better.	Yes, it is based in a sound rationale and it is a structured approach.
Are the results logical?	Yes, they are.	I agree with the percentages presented.
How is this model different than the one you currently use?	Integration is currently estimated as a standard on-top to SW and HW D&D effort percentage.	There is currently no approach for Integration effort estimation. However, this approach gave us better visibility and understanding of the Integration issue.
Is there anything not covered from the suggested approach?	If there were also values for the subcategories, this would enhance the quality of the approach even more.	Integration depends on factors that can not be estimated before hand. Therefore, this approach is the only way forward.
Any additional comment?	The integration sub-categories percentages can not be predicted in a general manner, since it is a case by case issue.	As Expert 1

8.2.3.1. Presentation of the results

As it can be observed from the expert's answers, there is no comment against the rational or the applicability of the suggested methodology, therefore, the suggested methodology is considered validated.

8.3. Summary

In this chapter, the third and fourth parts of the ES D&D effort estimation framework, the Reuse and the Integration modules, are presented. For the Reuse module, reuse matrices were developed, according to if the ES is created by a new or existing supplier and if it is going to be used in a new or an existing platform.

These Reuse matrices offer reuse percentages for each ES and for each of the potential situations it can be found into. For Integration, a 'percentage' approach was used. These percentages were derived by experts' knowledge and provided wider visibility into the Integration issue.

In the next chapter, chapter 8, an integrated view of the complete ES D&D effort estimation module and its validation by an additional 2 automotive OEMs are presented.

9. Framework Implementation and Validation

In the previous Chapters (6, 7 and 8), the development of an ES D&D effort estimation framework was presented. The individual modules were validated through interviews with automotive industry experts.

In order to be tested for its portability (to other OEMs) and its accuracy, the model was validated through workshops held with 2 additional OEMs. In this chapter, the complete ES effort estimation framework, as well as the results of its presentation with the 2 OEMs are being presented.

9.1. Framework Presentation

Using the HW, SW, Reuse and Integration modules as they were presented in chapters 5 to 7, individual values for HW, SW, Reuse and Integration are obtained. These values represent the required amount of D&D effort for each of the ES D&D effort domains. However, these values have to be combined together in order for the overall ES D&D effort to be estimated. The individual ES D&D efforts are combined together using the ES D&D effort distribution percentages as these were derived by the experts in the D&D effort capture workshop (see chapter 5). In this workshop, it was derived that for an item designed from scratch, the effort distribution is 14% on HW, 31% on SW and the rest 55% in the Integration.

Therefore, the overall ES D&D effort is estimated as follows: in the first step, the HW and the SW D&D efforts are estimated using the procedures described in chapters 5 and 6 respectively. After HW and SW D&D efforts have been estimated, amended HW and SW D&D efforts are derived by applying the appropriate amount of HW and SW reuse (as this was described in chapter 8) to the original HW and SW D&D effort. At the end, the overall ES D&D effort is estimated as [(HW amended –by Reuse- Effort)+(SW amended –by Reuse- Effort)]+(Integration Effort). Integration effort accounts for the 55% of the overall ES D&D effort (Effort capture workshop, chapter 5), and can easily be estimated either from the HW or the SW effort, since HW D&D and SW D&D effort is 14% and 31% of the overall ES D&D effort respectively.

The above-described process is automated and implemented in Microsoft Excel. In Figure 9.1, the initial screen of the framework is being presented. It contains 5 option buttons (HW D&D Effort Estimation, SW D&D Effort Estimation with Use Cases, SW D&D Effort Estimation with Statecharts, Reuse Percentage estimation and Integration Effort Estimation) and 5 respective text boxes, where the value for each corresponding module is displayed. At the end, there is a text box titled ''Embedded Systems D&D Effort'', where the overall ES D&D effort, as it is derived by the combination of HW, SW, Reuse and Integration modules is being displayed.

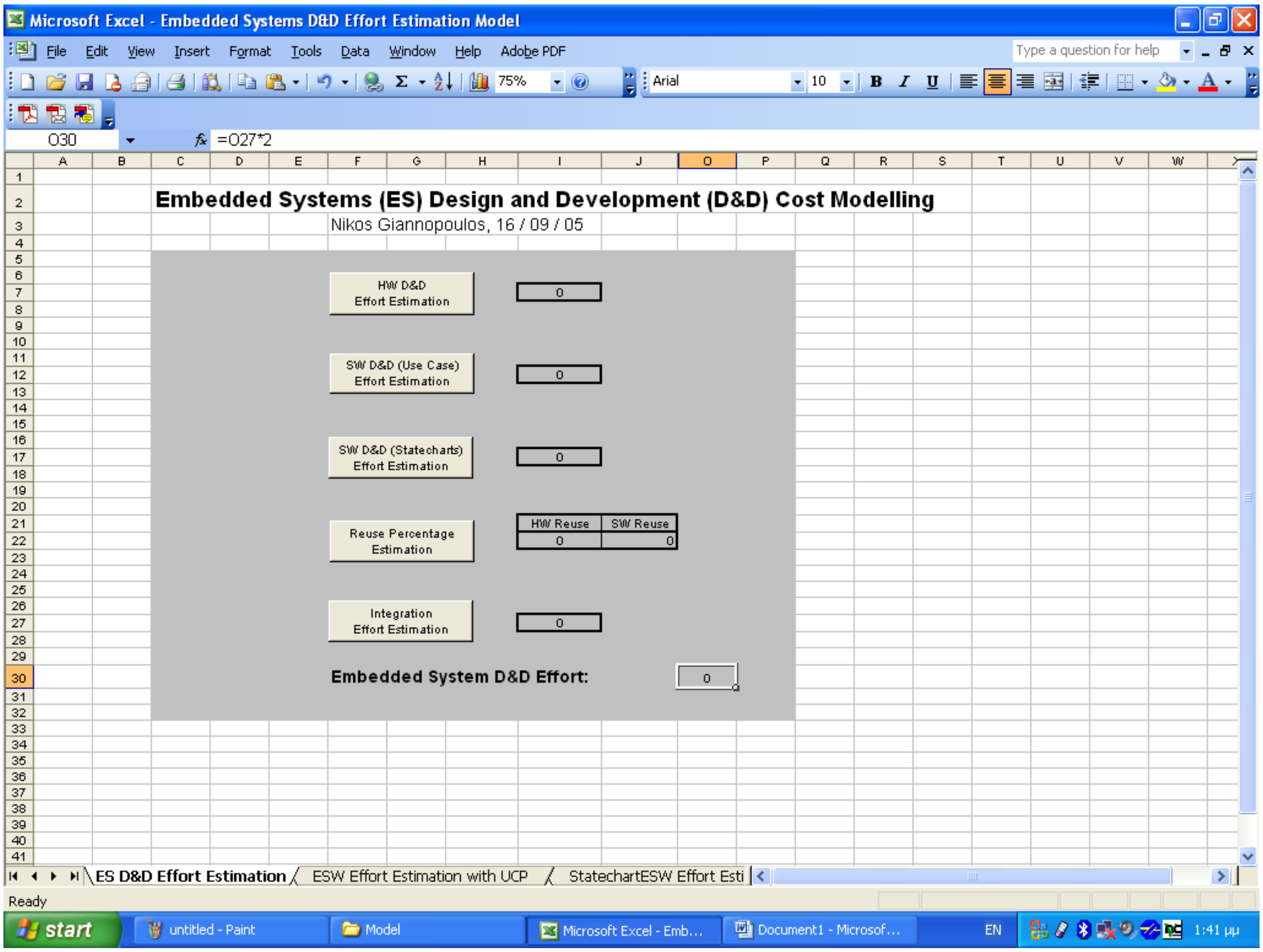


Figure 9.1: ES D&D Effort Estimation Framework Initial Screen

From the initial screen, the user can be transferred to any of the corresponding effort estimation modules, by simply clicking on the corresponding option button. If, for example, the user clicks on the ''HW D&D Effort Estimation'' option button, he will be transferred to the HW D&D Effort Estimation screen (Figure 9.2), which he can access the HW D&D Effort Estimation module through and perform an estimation for the ES HW D&D effort.

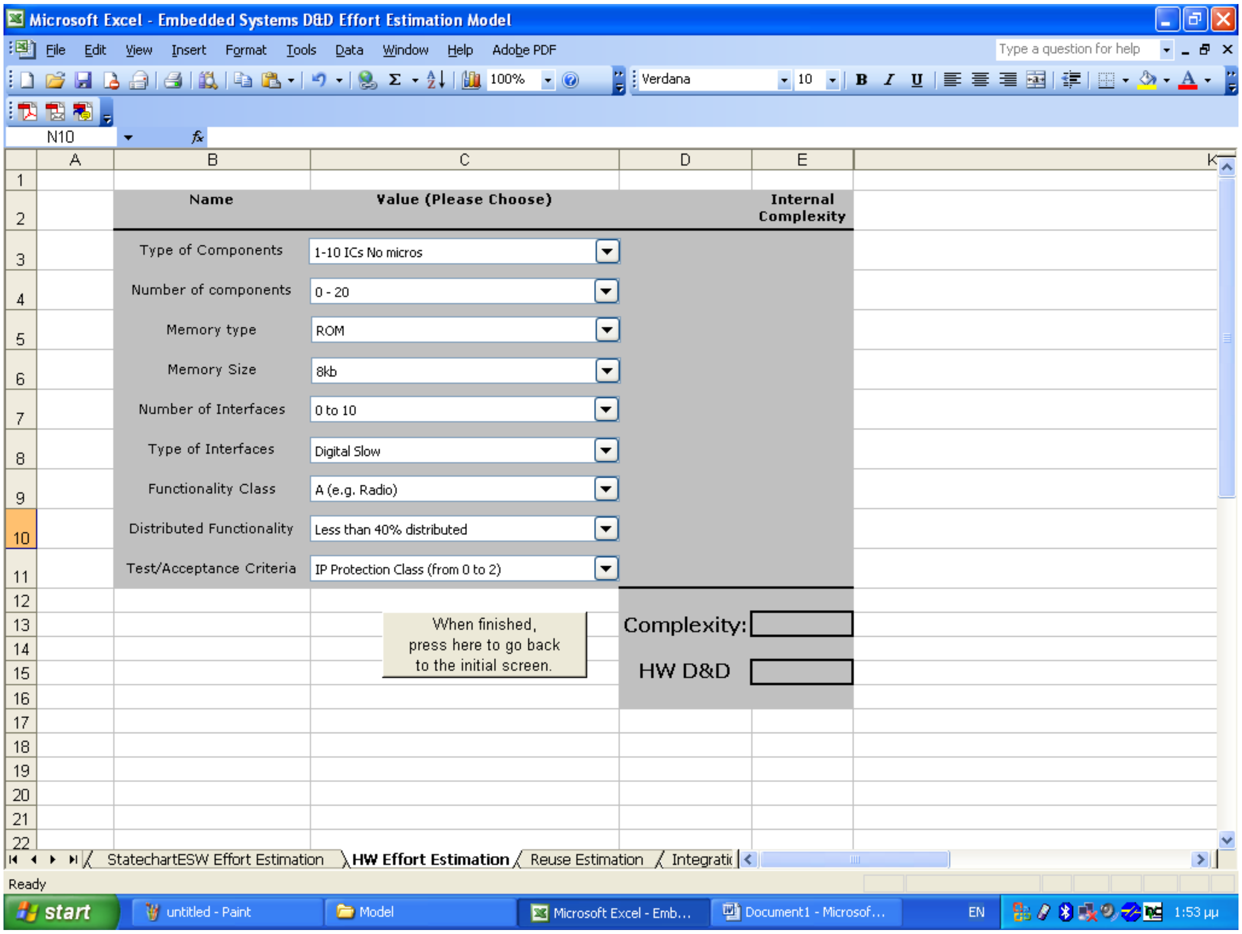


Figure 9.2:Initial Screen of the HW D&D Effort Estimation Module:An Example

Each of the drop down menus in this module (Figure 9.2) contains the corresponding HW complexity subcategories, as they were defined and presented in chapter 5. The user chooses the right HW D&D complexity factor subcategory and the program calculates the "Internal Complexity", the overall Complexity and the HW D&D Effort (Figure 9.3) using the process described in chapter 5. When HW D&D Effort has been derived, then the user clicks on the ''When finished, press here to go back to the initial screen" button and he is redirected to the initial screen, where the HW D&D effort value has been automatically conveyed by the program to its corresponding cell.

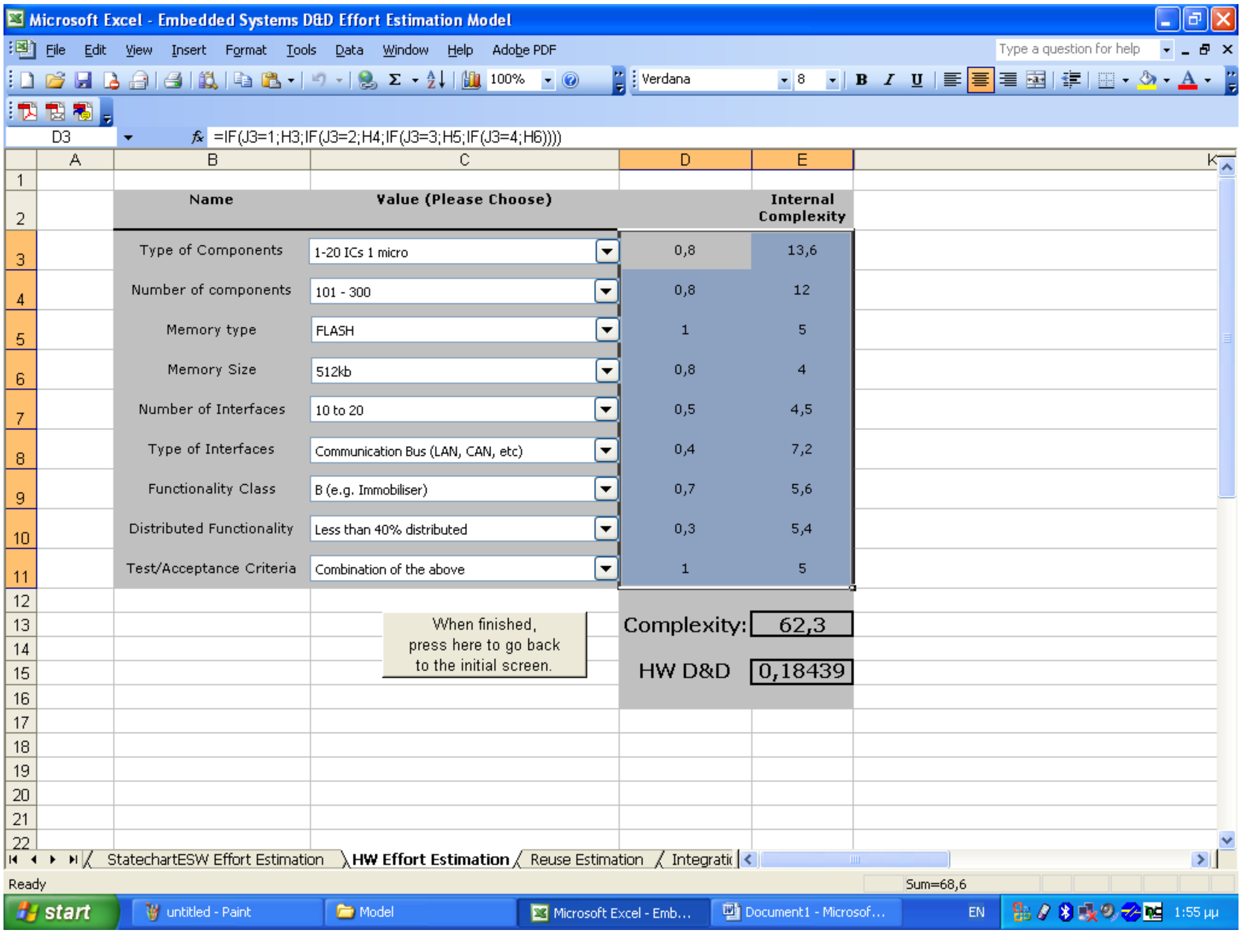


Figure 9.3: HW D&D Effort Estimation Module After Introducing Values:An Example

The same procedure is followed for the estimation of the embedded SW D&D effort. The user, according to what embedded SW case he has to estimate (Use Case or Statecharts) he presses the synonymous button on the initial screen and he is transferred to the corresponding page of the model (for Use Cases: screen shown in figure 8.4, whereas for Statecharts: screen shown in figure 9.5). There, the user enters the appropriate values and the embedded SW effort is calculated: in case of the UCP, he enters the counts for the actors, the use cases, the environmental factors, the AUCP of the known project and the D&D effort of the known project (figure 9.6), whereas in the case of Statechart based SW, he enters the counts for each of the complexity factors, the ACSM of te known project and the D&D effort of the known project (figure 9.7). In figure 8.6, the researcher has entered the values of SW case study 4 (as it was presented in the example at chapter 6, page 10), whereas at figure 9.7 the researcher has entered the values of SW case study 1, as it was presented as an example at chapter 7). When the SW estimation is over, the user clicks on the ‘’When finished, press here to go back to the initial screen’’ button and he is redirected to the initial screen. From there, he can continue with estimating Reuse (figure 9.8).

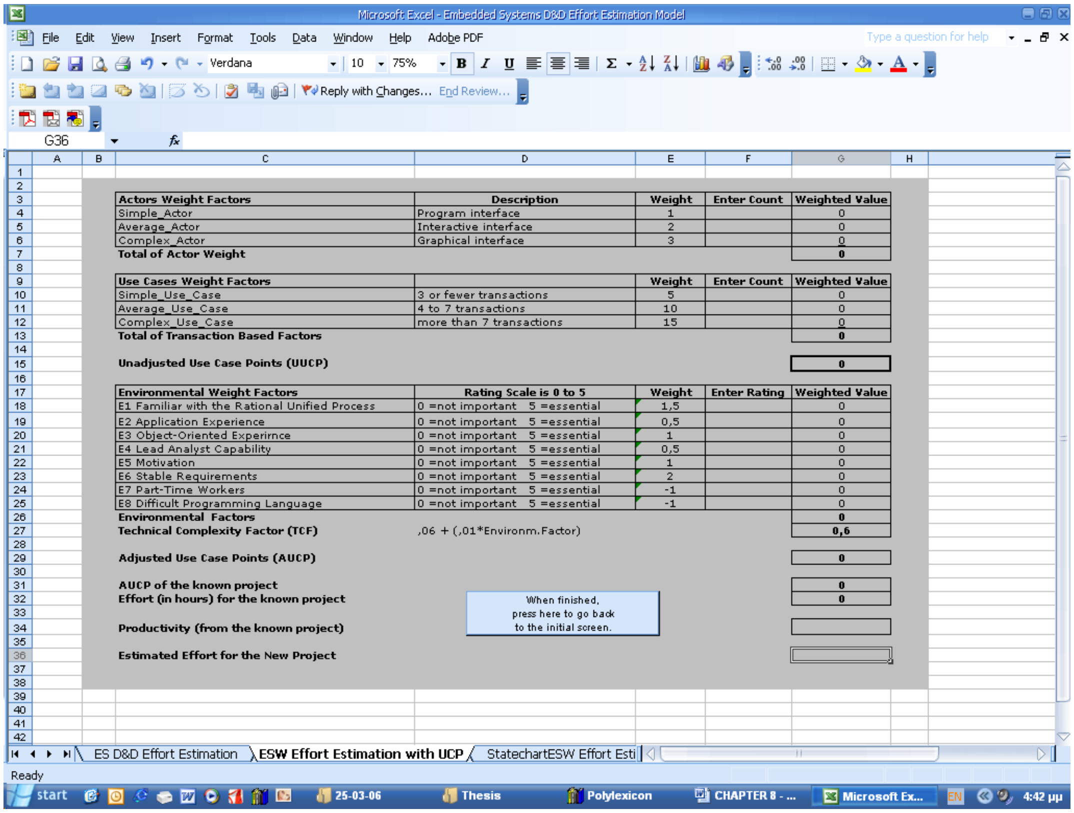


Figure 9.4: SW D&D Effort Estimation Module (Use Case Based)

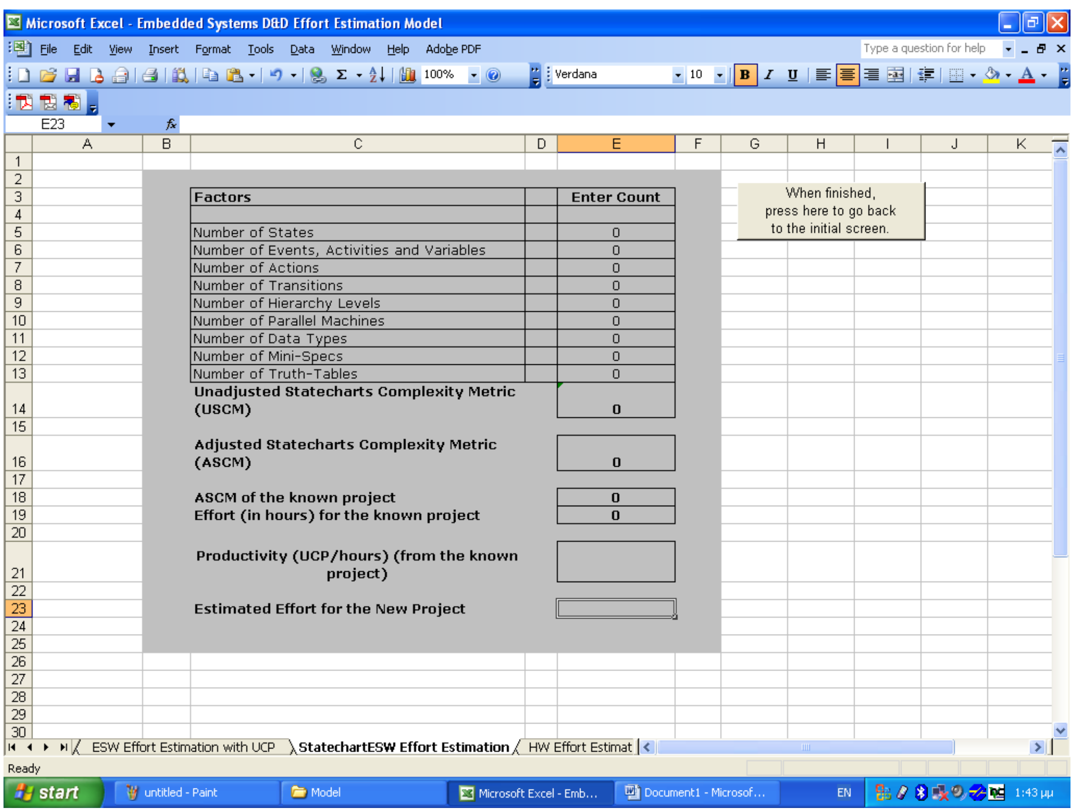
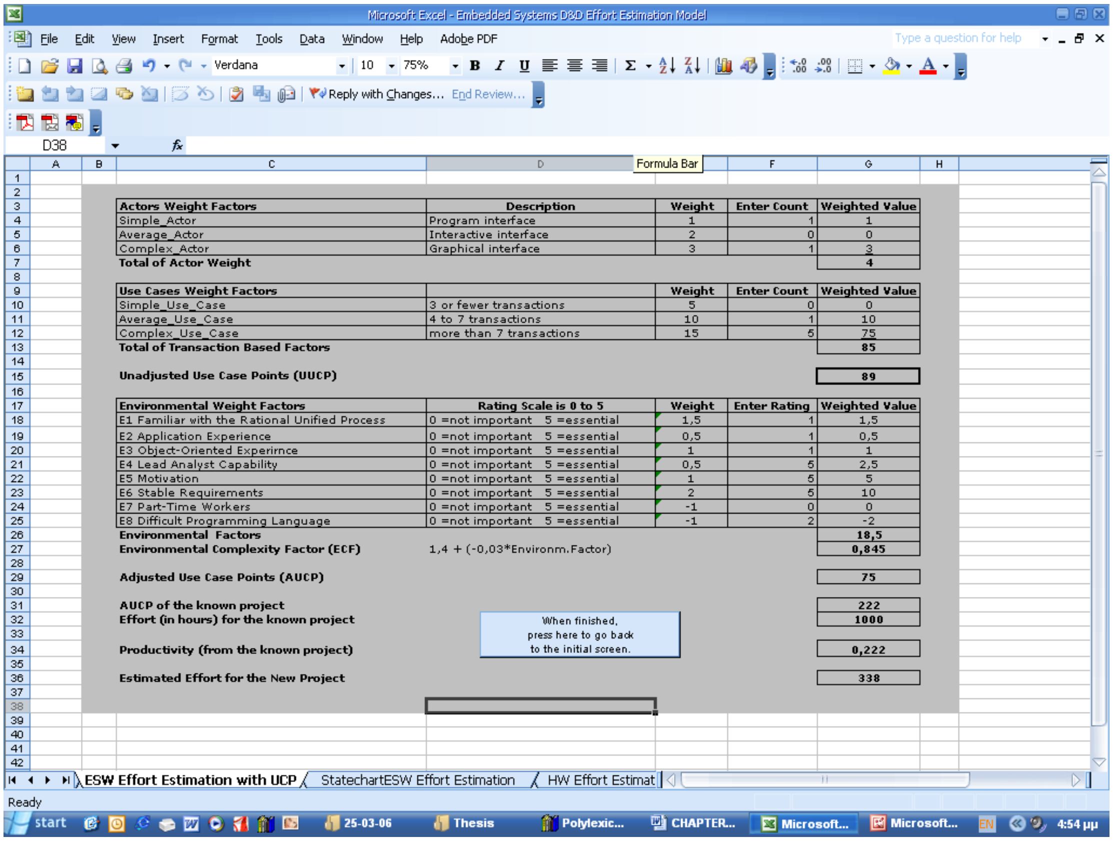


Figure 9.5: SW D&D Effort Estimation Module (Statecharts Based)

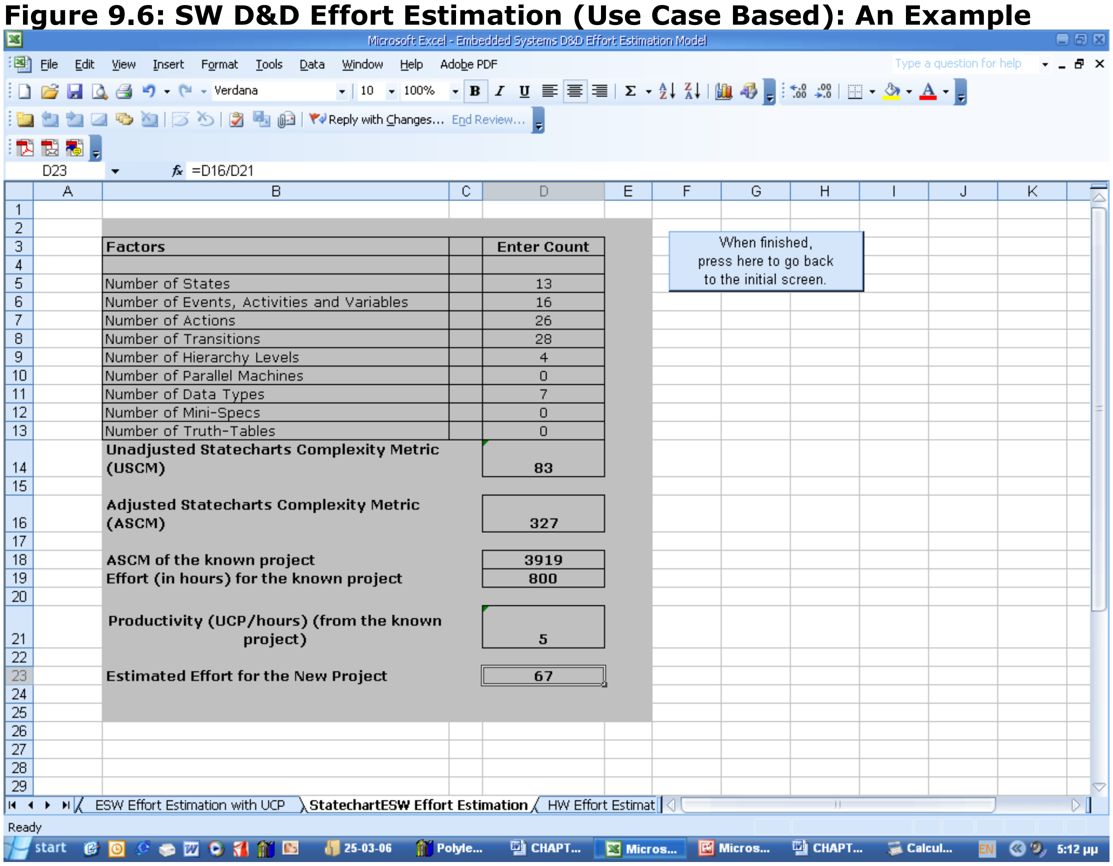


Actors Weight Factors	Description	Weight	Enter Count	Weighted Value
Simple_Actor	Program interface	1	1	1
Average_Actor	Interactive interface	2	0	0
Complex_Actor	Graphical interface	3	1	3
Total of Actor Weight				4

Use Cases Weight Factors		Weight	Enter Count	Weighted Value
Simple_Use_Case	3 or fewer transactions	5	0	0
Average_Use_Case	4 to 7 transactions	10	1	10
Complex_Use_Case	more than 7 transactions	15	5	75
Total of Transaction Based Factors				85
Unadjusted Use Case Points (UUCP)				89

Environmental Weight Factors	Rating Scale is 0 to 5	Weight	Enter Rating	Weighted Value
E1 Familiar with the Rational Unified Process	0 =not important 5 =essential	1,5	1	1,5
E2 Application Experience	0 =not important 5 =essential	0,5	1	0,5
E3 Object-Oriented Experirnce	0 =not important 5 =essential	1	1	1
E4 Lead Analyst Capability	0 =not important 5 =essential	0,5	5	2,5
E5 Motivation	0 =not important 5 =essential	1	5	5
E6 Stable Requirements	0 =not important 5 =essential	2	5	10
E7 Part-Time Workers	0 =not important 5 =essential	-1	0	0
E8 Difficult Programming Language	0 =not important 5 =essential	-1	2	-2
Environmental Factors				18,5
Environmental Complexity Factor (ECF)	1,4 + (-0,03*Environm.Factor)			0,845
Adjusted Use Case Points (AUCP)				75
AUCP of the known project				222
Effort (in hours) for the known project				1000
Productivity (from the known project)				0,222
Estimated Effort for the New Project				338

Figure 9.6: SW D&D Effort Estimation (Use Case Based): An Example



Factors	Enter Count
Number of States	13
Number of Events, Activities and Variables	16
Number of Actions	26
Number of Transitions	28
Number of Hierarchy Levels	4
Number of Parallel Machines	0
Number of Data Types	7
Number of Mini-Specs	0
Number of Truth-Tables	0
Unadjusted Statecharts Complexity Metric (USCM)	83
Adjusted Statecharts Complexity Metric (ASCM)	327
ASCM of the known project	3919
Effort (in hours) for the known project	800
Productivity (UCP/hours) (from the known project)	5
Estimated Effort for the New Project	67

Figure 9.7: SW D&D Effort Estimation (Statecharts Based): An Example

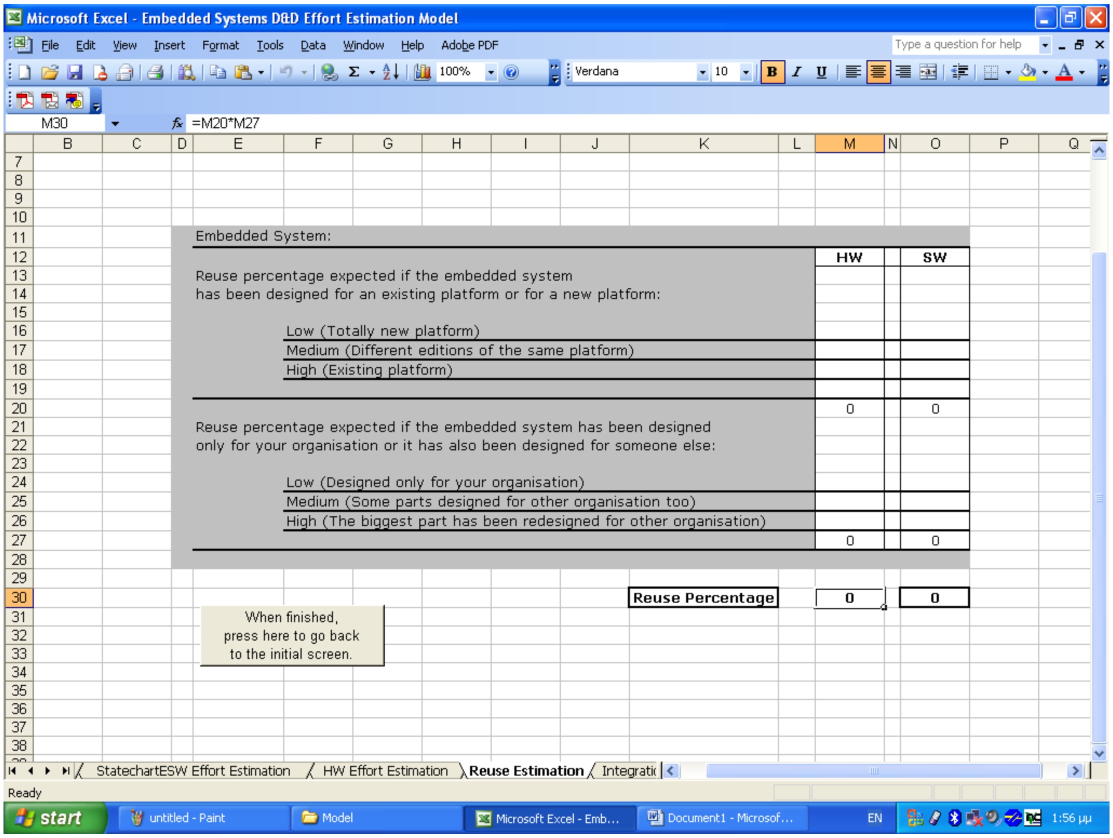


Figure 9.8: Reuse Effort Estimation Module

The above picture is the initial picture of the Reuse part of the model. In this screen, there are no initial values, since these (as it was explained earlier in chapter 8) will depend, each time, by the system under examination. If the system under examination has been used already in another estimation, then its reuse percentage matrix would be retrieved by the database, whereas if it is the first time this system is estimated, then the expert would assign the necessary values. For example, in figure 9.9, the case that a system whose reuse percentage matrix has already been created it is displayed. When the Reuse estimation is over, the user clicks on the ''When finished, press here to go back to the initial screen'' button and he is redirected to the initial screen, where he can proceed with the Integration estimation.

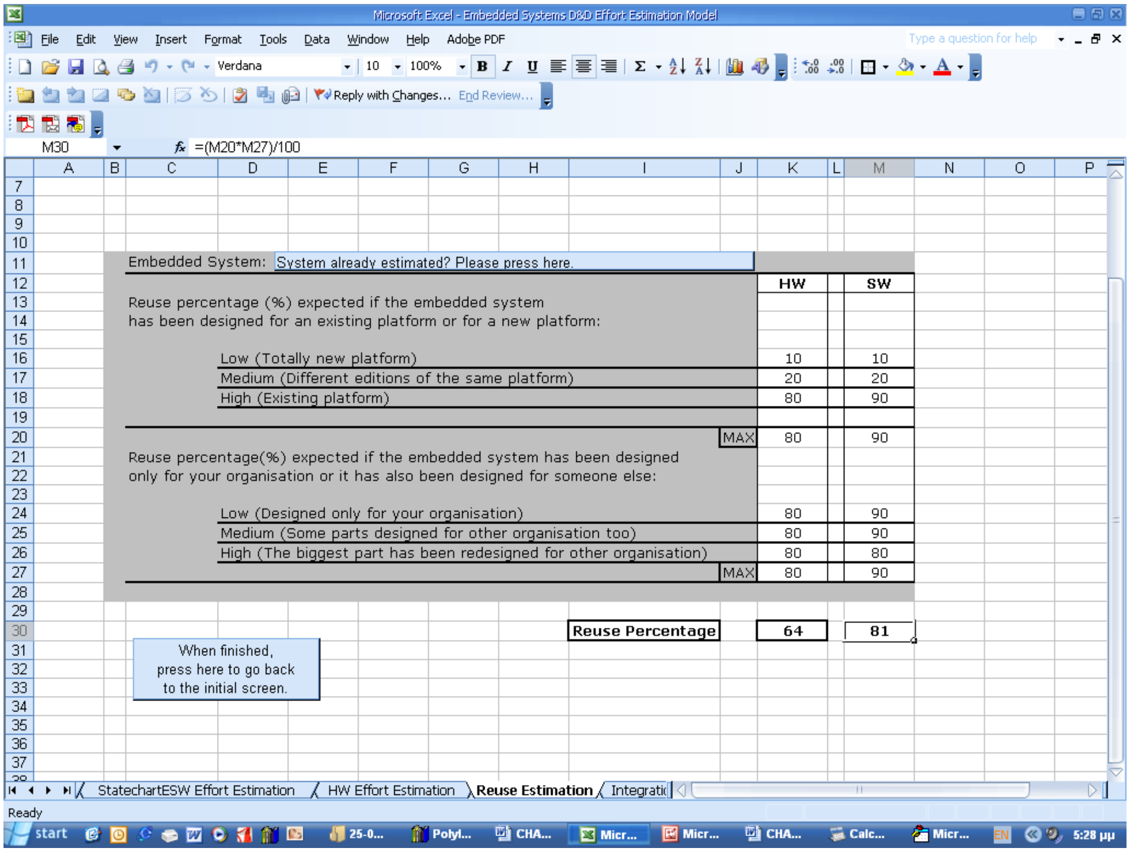


Figure 9.9: Reuse Effort Estimation

The following figure presents the Integration part screen. In this screen, the percentages are already set, since, as it was derived at chapter 8, these percentages are generic for these categories for the automotive industry.

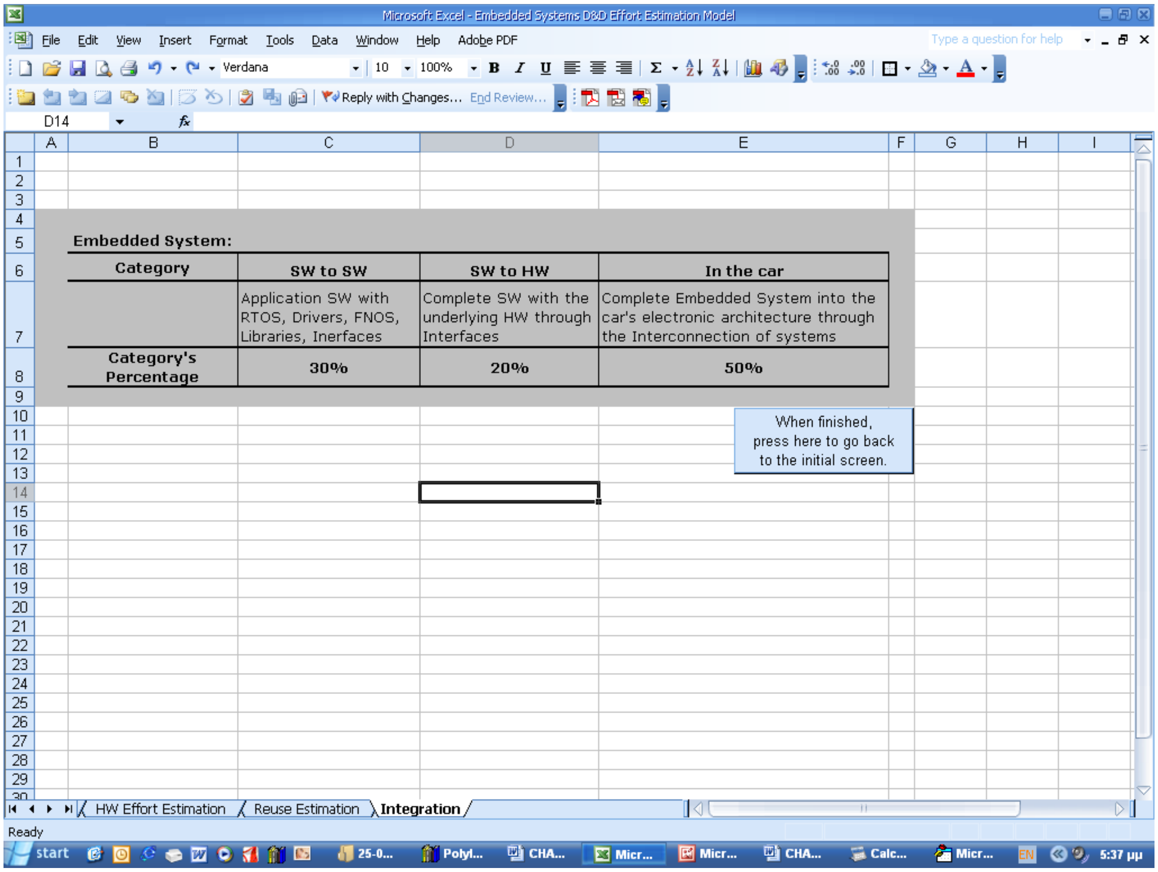


Figure 9.10: Integration Effort Estimation Module

When all the individual efforts have been estimated and conveyed to the initial screen, the overall ES D&D effort is estimated as described earlier: [(HW amended –by Reuse- Effort)+(SW amended –by Reuse- Effort)]+(Integration Effort). The value of the overall ES D&D effort is displayed in the 'Embedded Systems D&D Effort' on the initial screen. Amended HW and SW efforts are calculated by multiplying the HW or SW effort estimated by the corresponding framework's module, times the corresponding Reuse percentage.

In the previous pages, the framework the researcher developed was displayed in its computerised edition, as it is implemented in the excel spreadsheet. Although the framework's presentation was done module by module, the examples used for each module's presentation were independent of each other, as it was not possible to acquire a complete case study (a case study where values would exist for each of the framework modules). Therefore, the researcher had to follow the above displayed way of presenting the estimation framework.

9.2. Framework Validation

In order to assess the framework for its logic and portability, the researcher organised 2 workshops with 2 additional OEMs. Two individual workshops, which lasted for 2 hours each, were held in each of the OEMs' premises.

9.2.1. Workshop Design

9.2.1.1. Target Audience

3 experts from the first OEM and 2 experts from the second OEM participated in the 2 individual workshops respectively. These experts, although of various skills (ie product engineer, purchaser, etc), they were selected from the interviewed organisations to be present in the workshop in order to provide a more in-depth and broader view on the validation process. A complete list with the participants' job roles, and years of experience can be found in tables 1 and 2 bellow:

Table 9.1: First OEM Workshop Participants

Name	Job Role	Experience
Expert 1	Electrical Research & Vehicle Technology	6 years
Expert 2	Project Leader, Systems Engineering	6 years
Expert 3	Electronics Cost Estimator	4 years

Table 9.2: Second OEM Workshop Participants

Name	Job Role	Experience
Expert 1	Senior Applications Engineer, Elctronics&Electrical Engineering	5 years
Expert 2	Cost Estimator	10 years

Each module was validated separately, because none of the 2 OEMs had a complete ES case study (specifications and data for each of the ES modules for the same ES), which could be used to validate the framework as a whole. The validation of the framework was based on the results achieved with the sponsoring organisation's data, since none of the 2 OEMs had case studies (own case studies) available even for each module separately. In addition, due to time restrictions and physical unavailability, SW experts from both OEMs was not possible to be present, and therefore the SW module was not possible to be validated by both OEMs in these workshop.

For each of the modules, the same with the sponsoring organisation's validation questionnaires were used (for HW, Questionnaire G, for Reuse, Questionnaire J, and, for Integration, Questionnaire K) in order to ensure consistency and be able to create a common base of reference on comparing and analysing the results. Therefore, the workshops' objectives were:

1. Explain the framework's rationale
2. Explain, for each module, its rationale, the data it requires and the way it works
3. Validate the framework modules

9.2.1.2. Conducting the Interviews

Both workshops were carried out following the procedure described bellow: Initially, the researcher presented to the experts an introduction to the scope, the aim and the objectives of the workshop, in order to 'set the scene' and familiarise the experts with the procedure. After this initial stage, the workshop was initiated with the use of the semi-structured questionnaires.

Since the SW related module was not possible to be assessed during these workshops, the 2 hours workshop time was divided in 4 periods: 30 minutes for the HW module validation, 30 minutes for the Reuse module validation, 30 minutes for the Integration module validation, and at the end, 20 minutes for general discussion and comments. This was also the sequence the individual modules were examined to be validated.

The validation of each module was performed as follows: firstly, the researcher explained to the experts the rationale of each module, the data it requires and the way it works and in the second stage, the validation of the module was performed by the experts using the appropriate corresponding questionnaire. After the validation of all the modules, at the end of the workshop, there was a short presentation of the computerised edition of the framework and an open discussion about the framework in general.

Both workshops were captured on tape and paper, to ensure accuracy on interpretation and analysis of results, as well as a point of reference when a doubt occurs. A summary of the experts' answers is presented in the next pages:

Table 9.3: OEM 1 Experts Opinions' Summary for HW Validation

Questions	Expert 1	Expert 2	Expert 3
Do you agree with the rational of the suggested framework?	Yes, the rationale is sound.	Yes, the module's rationale is sound and clear.	Yes, I do.
Are the results logical?	R^2 indicates they are.	Yes, they are.	It seems that there is a good degree of correlation.
Could the suggested framework be used to predict HW D&D cost based on the HW complexity?	Yes, it could.	Only for Body Control items.	For Body Control only.
Is there anything not covered from the model?	Infotainment and Chassis	Infotainment and Chassis	Infotainment and Chassis
How is this model different than the one you currently use?	An internal tool is currently used.	A budget is set, based on our understanding and supported by the internal tool.	A budget is set, based on past data for the specific functionality and supported by the internal tool.
What are the potential issues in implementing such a methodology?	More data have to be collected for further calibration of the model.	The fact that on the specification stage it is very difficult to judge the complexity factors' subcategories.	The data to be collected and the difficulty on assigning values.

Table 9.4: OEM 1 Experts Opinions' Summary for Reuse Validation

Questions	Expert 1	Expert 2	Expert 3
Do you agree with the rational of the suggested framework?	Yes, I do.	Yes, the rationale is sound.	Yes, the module follows a sound rationale.
Are the results logical?	Yes, they are. However, they are company-specific.	The results seem logical.	They seem to be. These percentages are company and supplier specific.
Could the suggested framework be used to predict Reuse (for both HW and SW)?	Yes, it could.	The suggested approach could support the estimation of HW and SW reuse.	Yes, it could.
Is there anything not covered from the model?	No, the model is complete	The model seems complete.	It is complete, for the specific company and supplier.
How is this model different than the one you currently use?	There is currently no model used for reuse estimation within the organisation.	No model is currently used to predict Reuse.	Reuse is not currently estimated within our organisation.
What are the potential issues in implementing such a methodology?	Data (reuse percentages) have to be collected. Reuse tables could also be created for each individual supplier.	Data (reuse percentages) have to be collected by the engineers.	The data that have to be collected.

Table 9.5: OEM 1 Experts Opinions' Summary for Integration Validation

Questions	Expert 1	Expert 2	Expert 3
Do you agree with the rational of the suggested approach?	It is a logical approach on the integration issue.	Yes, I agree.	It is a structured approach on confronting the integration issue.
Are the results logical?	They seem to be.	It is a case-by-case issue.	They depend on the individual supplier and the item itself.
How is this model different than the one you currently use?	Integration is currently estimated based on each individual case.	Integration is currently estimated on an individual basis, based on the item and the supplier.	On a case-by-case basis, taking into account the item's supplier and the item itself.
Is there anything not covered from the suggested approach?	If there were also values for the subcategories, this would enhance the quality of the approach even more.	Integration depends on factors that cannot be estimated before hand. Therefore, this approach is the only way forward.	Integration depends on factors that cannot be estimated before hand. For us, it is a case-by-case issue.
Any additional comment?	---------	---------	---------

Table 9.6: OEM 2 Experts Opinions' Summary for HW Validation

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested framework?	It is s logical and well reasoned approach	Yes, the module follows a logical sequence of steps.
Are the results logical?	The results seem logical.	Correlation coefficient R^2 shows they are.
Could the suggested framework be used to predict HW D&D cost based on the HW complexity?	Yes, it could, for items belonging to the body domain of the car.	Yes, it could.
Is there anything not covered from the model?	The areas of Infotainment and Chassis. The module does not also provide or technology changes, ie Bluetooth.	Infotainment and Chassis
How is this model different than the one you currently use?	No model is currently used. We use expert judgement based on our past experience.	No model is currently used to predict HW D&D effort.
What are the potential issues in implementing such a methodology?	The data that have to be collected plus potential problems that could occur if the module produces very different output than the supplier's. In that case, the supplier could withdraw and the car line could remain unfulfilled.	Data collection. More than 6 case studies are necessary for coefficients' calibration.

Table 9.7: OEM 2 Experts Opinions' Summary for Reuse Validation

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested framework?	Doesn't feel so logical as the others, because the supplier could –for the same ES and the same car- to introduce a new version of the ES by choosing (ie) a new micro or a new PCB.	Yes, the model follows a sound structure.
Are the results logical?	They seem to be, but they are company and supplier specific and depend on the technology advancements.	The results seem logical, they are however depended on each company.
Could the suggested framework be used to predict Reuse (for both HW and SW)?	Yes, it could, if we are sure no change (see answers in the previous questions) has been done to the ES due to technological advancements.	The suggested approach could support the estimation of HW and SW reuse.
Is there anything not covered from the model?	It does not take into account technology advancements.	As Expert 1.
How is this model different than the one you currently use?	There is currently no model used for reuse estimation within the organisation.	No model is currently used to predict Reuse.
What are the potential issues in implementing such a methodology?	Data (reuse percentages) have to be collected. Reuse tables could also be created for each individual supplier.	Data (reuse percentages) have to be collected by the engineers.

Table 9.8: OEM 2 Experts Opinions' Summary for Integration Validation

Questions	Expert 1	Expert 2
Do you agree with the rational of the suggested approach?	It is s logical and well reasoned approach	Yes, the module follows a logical sequence of steps.
Are the results logical?	Yes, they seem to be. I would assign SW/SW 15%, SW/HW 25% and ''In the car'' 60%. However, I would break-down the ''In the car'' integration subcategory into the ''In the car'' and the ''Integration of the signals from outside'' like (ie) radio or Bluetooth or mobile phones signal categories. Then, the percentages are as follows: SW/SW 15%, SW/HW 25%, and ''In the car''/''Outside signals'' 30%/30% or 20%/40% (it depends on the case). This is more valid for the infotainment domain.	I agree with the percentages presented.
How is this model different than the one you currently use?	No model for Integration estimation is currently used.	There is currently no approach for Integration effort estimation.
Is there anything not covered from the suggested approach?	If there were also values for the subcategories, this would enhance the quality of the approach even more.	Integration depends on factors that cannot be estimated before hand.
Any additional comment?	The integration sub-categories percentages cannot be predicted in a general manner, since it is a case by case issue.	As Expert 1

9.2.2. Results Presentation

Having the information collected, the results were analysed by examining the answers given by the experts from both OEMs, as well as by analysing the discussions that took place during the workshops. A comparison with the answers given by the sponsoring organisation's experts (paragraphs 6.3.4, 7.4.1.4, 8.1.4 and 8.2.4) was also performed. The researcher looked for consensus between the answers, as well as for commonalities, differences and specific to each OEM cases.

From what can be observed from all the answers (OEM 1, OEM 2, sponsoring organisation) the three modules (and therefore the framework) are based in sound rational, they work in structured logic and produce logical results. In addition, all the modules can be used to estimate ES D&D (HW, SW, Reuse and Integration) effort; in fact, they constitute a step forward on structuring, supporting and rationalising the ES effort process, especially since all the organisations involved (OEM 1, OEM 2 and the sponsoring organisation, plus the organisations that participated in the AS-IS study on chapter 4) use expert judgment to derive these values.

A general observation made by all the participants is that although each of the framework's modules has been developed using case studies from different car domains (ie for SW, case studies from the infotainment domain were used, whereas for HW, case studies from Body domain), the framework could be expanded to cover the rest of the domains, if additional data collection and analysis is performed. Looking in more detail into each of the individual modules, the following key observations can be obtained:

- **HW D&D:** the current module has been designed for the car Body domain only. Therefore, it does not cover the Chassis and the Infotainment area. In addition, data collection (from the experts) has to be performed, both for module calibration and for extending the module into covering the Chassis and Infotainment areas.
- **Reuse:** The Reuse percentages are company and supplier specific. This means that the percentages the sponsoring organisation provided do not necessarily apply to the 2 OEMs. Therefore, further data collection has to be performed, for ''Reuse Tables'' to be designed both based on supplier and on item.
- **Integration:** both OEMs and the sponsoring organisation agree that percentages into Integration subcategories cannot be assigned, since they

are a case-by-case issue. There is a small variation regarding the percentages of the main Integration categories. The first OEM noted that although the sponsoring organisation's percentages seem logical, it could not provide its own, since its Integration case is a case-by-case issue and the percentages could vary (not significantly though). OEM 2 gave percentages (for the main Integration categories) that are very close to the percentages the sponsoring organisation provided. OEM 2, in the case of Integrating an Infotainment module into the car, broke down the ''In the car'' Integration category into two categories, ''In the car'' and the ''Integration of the signals from outside'' categories.

Again, the percentages assigned to these two categories (30%/30% or 20%/40%) sum up to the 60% provided as percentage assigned to the ''In the car'' Integration category when a Body item is considered. Nevertheless, this structured approach is a step forward for people to ask the right questions and derive company-specific percentages.

As it has become obvious from the experts' answers, Reuse and Integration are case specific issues, and it is not appropriate for generic percentages or numbers to be assigned to. The best the expert can in each case do is to identify the factors that may affect the two areas so he can assign the appropriate percentage or number for each individual case. However, this study has improved understanding about benefits for Reuse and Integration cases.

Since there was no objection against the rationale, the structure or the operation of the framework's modules, the modules presented are considered validated.

9.3. Summary

In this chapter, the author presented the complete ES D&D effort estimation framework in an integrated form and described how the individual effort estimation modules work together in order for an estimation to be performed. The whole estimation process has been automated using Microsoft© Excel©. In addition, in this chapter the validation of the framework by an additional 2 automotive OEMs was presented. The results were validated and accepted by all the participants of the study. In the next Chapter, chapter 9, the whole thesis is discussed and the thesis conclusions are drawn.

Chapter 10: Discussion and Conclusions

10.1. Introduction

In chapters 5 to 8, the individual modules of the ES D&D effort estimation framework were presented. To be tested for its generalisability, the complete framework was presented and validated by two additional OEMs. This, along with the validation results, was presented in chapter 8. In this chapter, the implications occurred by these findings are discussed and conclusions are derived.

10.2. Key Research Observations

10.2.1. Literature Review

The research is a multidisciplinary one, since it involves the areas of ES (in general), HW, SW, Reuse and Integration (as individual areas) as well as the area of Cost prediction. Therefore, the researcher had to thoroughly investigate all the above aforementioned areas.

Literature review established the significance of estimating the costs of designing and developing an ES. In addition, evidence was presented regarding the importance an accurate ES D&D cost estimation bears for an organization involved in high mass production, like in the automotive industry.

OEMs outsource the majority of their ES. Specifications are produced and then passed to the suppliers, who are then responsible of producing the ES with the desired functionality. Although a very limited number of approaches and/or frameworks have been suggested for estimating the D&D effort of an ES (described in Chapter 2), none of them could be applied by the OEM in the specification stage because:

1. each implementation (the physical ES realization) can be accomplished through a variety of possible alternative HW/SW combinations,
2. existing approaches and/or frameworks for ES D&D effort estimation cover only an ES part (SW or HW, not both, and most of them do not cover the integration phase), and
3. implementations are protected by supplier's IP rights, therefore, necessary for any estimation information (ie LOC, number of ICs, etc) are not disclosed to the OEM, preventing –this way- the deployment of any formal estimating method.

Another issue comes from the fact that each ES is developed by a different supplier as a stand-alone item, separately from the remaining car's electronic architecture. In addition, various suppliers exist in any OEM's supply chain. This leads to unpredictable side effects when the designed by the different suppliers systems are interconnected, requiring extensive testing and incurring additional costs for the necessary changes to be implemented in order for the ES to operate as it must. However, because these costs incur under these circumstances make them unpredictable before hand; this is the reason why expert judgment (EJ) is still the most commonly cost estimating method used within OEMs when estimating ES D&D effort (and therefore cost) in the specifications stage.

The author observed a lack of research in the area of ES D&D cost modeling and practices. Although new approaches –like platform based designed, modeling the complete car's functionality from the beginning, OEM and supplier codesigning and codeveloping the ES, etc- have been introduced as means for controlling the ES D&D cost, these approaches are still developing or not used to their full extend. This seems to be a generic trend, since the same situation is observed in the D&D effort estimation in general (non ES), where only a very limited number of models/frameworks have been developed for D&D effort prediction. The reason for this is that D&D depends on factors that are very difficult to be quantified (ie the ability of the engineer to evaluate alternative solutions, etc).

Therefore, the author concentrated his efforts in developing a framework for linking the ES specifications with the actual ES implementation, bypassing that way the ''supplier's blockage'' (the situation that necessary to OEM information in order to perform his estimate is not available because it is protected by supplier's IP rights). To accomplish this, the author based his approach in the exploitation of the ES specifications since this is the only information OEM holds in the specifications stage. It was therefore important to investigate if any relationships between the early design stages, the specifications, the ES implementation and its cost exist, and if yes, if these relationships could be used for ES D&D cost prediction.

10.2.2. AS-IS

The key benefit from the AS-IS study was to validate what was observed during literature review: that there exists the need for improving the ES D&D effort estimation process. For this purpose, a workshop was organized –firstly- within the sponsoring organization, where, using the $IDEF_0$ standard- detailed documentation of the ES D&D effort estimation process the sponsoring organization follows was performed. This, gave the researcher the opportunity to identify the obstacles and the problems surrounding the overall ES D&D effort estimation process.

In a second stage, similar workshops for capturing the AS-IS of the ES D&D cost estimation process were held in an additional 3 organisations (1 from automotive and 2 from the aerospace/defence industry) in order for commonalties and differences to be identified firstly within industries and secondly, across them. However, due to people unavailability and time restrictions it was not possible for detailed AS-IS documentation with the use of the $IDEF_0$ standard to be performed.

AS-IS study shown that ES D&D effort estimation is a very important issue for all companies regardless of their industry sector. The major problem is the absence of information. Even in cases of companies that still have a -limited though- in house production of ES (having therefore the experience of past projects), estimating D&D cost of an outsourced ES proves extremely difficult. This is because no information on the system’s development is available to the estimator so he can perform an estimate. Only one company, of the aerospace/defence sector, was able to judge quotations with greater level of confidence. This was due to the reason that this company uses ES that change rarely, therefore knowledge from past projects can easily be applied.

Specific mention has to be made for embedded SW. In automotive industry, due to the increasing need for improving car’s efficiency and functionality, offering better value for money and reduced cost, changes in ES are very often. These changes are implemented in SW, because embedded SW is more flexible, brings no production cost and it is consider easy to change, especially in the late stages of the production process. In this late stage, even HW issues are addressed through SW changes. This is not the case in aerospace/defence, were the product is of a more stable nature.

The generic view, expressed by experts in all the four workshops was that considering what has been presented so far, there is the need for improving the ES D&D cost estimation process. This is in accordance with what was observed through literature review. In addition, AS-IS also validated the view (acquired through literature review) that integration is the major driver in ES D&D effort estimation, due to (i) the separate development of SW and HW and (ii) the interconnection in a common architecture of systems developed by different suppliers in isolation.

10.2.3. AS – IS versus TO – BE

After the AS – IS model had been derived, the researcher moved forward on creating the TO – BE model, based on opinions of experts. This TO – BE model became the basis for the developed by the researcher ES D&D effort estimation framework. The main observations of the AS – IS study where that:

1. There is no standardised practice/method neither for electronic items cost estimating, nor for estimating their D&D cost. In the contrary, each company follows its own approach/method.
2. Accessing information necessary for estimating the cost (manufacturing or even D&D) of an electronic item proves most of the times difficult, since this information are most of the times property of the supplier and they are not disclosed.
3. In the absence of these data, the estimator has to use Expert Judgement, which is the most popular estimating method across industry. Even in case information is available, Expert Judgement is utilised by the estimator when performing his estimation.
4. There is no differentiation between HW D&D and SW D&D when estimating the effort of D&D an electronic item; the SW part of the system is usually estimated as an overhead of the manufacturing cost.
5. The difficulty of accessing information due to the variety of reasons described in the previous paragraphs creates the need for a framework to link the product specifications with the cost estimating process, by-passing this way the '’supplier’s information blockage’’.
6. The method will need to be reusable so that cost engineers will be able to create estimates for D&D cost faster in the future for similar products.

TO – BE model, through its practical implementation, the newly derived effort estimation framework for embedded systems, addresses all these issues. The newly derived ES D&D effort estimation framework (an therefore the TO – BE model) offers a sound and well-established process for estimating the effort for D&D an embedded system. The framework explicitely differentiates between HW and SW efforts, provides for the estimation of Reuse and Integration and Testing and in addition, it takes into account, the Reuse and Integration and Testing effect towards the overall ES D&D effort estimation.

The framework does not depent on information blocked by the supplier; the required information can be easily extracted by the OEM’s specifications, and can be easily applied to the framework, limiting in this way the appliance of Expert Judgement by the estimator. As it was shown in the validation stage, the framework produces results that are very close to actual figures. At the end, the framework has been presented as a fully automated process, in Excel, making it that way easy to use, but most importantly, reusable, since every estimation can be saved for future reference.

10.2.4. Framework Development

Chapters 5 to 7 present the development of the ES D&D effort estimation framework. This framework confronts the issue of ES D&D effort estimation by directly correlating the system's specifications with the system's actual implementation or its actual development effort. A methodology was presented that allowed cost estimators to base their estimates on the functional characteristics of the product under investigation.

The framework created following a module by module concept. This was done for the following reasons: (i) this is the way interviewed organizations work and (ii) the corresponding departments within the organizations are disjointed with varying degree of available information. In other words, there was no case study that for the same item all the necessary specifications and information existed. Therefore, each module had to be developed and tested separately.

10.2.4.1. SW Module

The course followed by the researcher in the development of the SW module was common for both cases (UML Use Case or Statechart specifications): firstly, the complexity of the SW to be developed was assessed and then this complexity was used, together with the organisation's productivity, in order to derive the effort necessary to D&D the suggested embedded SW. In the case of specifications expressed in Use Cases, after evaluating all the alternatives, the Use Case Points method was applied in order for a complexity metric to be derived. In a second stage, this complexity metric was combined with the known SW D&D effort of a project in order for the productivity to be derived and, in a third stage, this productivity was utilized in combination with the complexity metric for the rest of the SW D&D efforts to be derived.

In the case of specifications expressed in Statecharts, after the evaluation of the alternatives, it became obvious that none of the available approaches could be applied to derive a complexity metric; therefore the researcher developed a new one. From this point forward, the course followed by the researcher in order to derive the unknown SW D&D efforts was the same with the one followed in the case of the UML Use Case specifications.

SW specifications may come in multiple ways, however the most observed in industry are in UML Use Cases, Statecharts and free text. In the case if this research, both UML Use Cases and Statecharts specifications were available. The researcher confronts the case of having specifications expressed in free text with creating the corresponding UML Use Case or Statechart diagram, otherwise the ‘’free-text’’ specification is very vague and open to misinterpretation and misunderstandings.

10.2.4.2. HW Module

The course followed by the researcher for the development of the HW D&D effort estimation was different to the one followed in SW D&D effort module development. Firstly, a metric for expressing the complexity of the indented HW was derived, and in a second stage, this complexity metric was correlated with the actual HW D&D cost of 6 HW implementations, creating a rule. From that point onwards, by applying this rule, the estimator was able to determine the HW D&D cost of a HW implementation.

The correlation coefficient of the obtained rule shown that there is a good amount of correlation within the rule (ie: between HW D&D complexity and HW D&D cost). However, the HW module was developed only for the body domain of the car, since it was possible to acquire case studies only from this domain. Although the rational of the framework could easily be transferred to the rest of the car domains (chassis, infotainment), the framework’s applicability to these domains should be re-examined when appropriate case studies are obtained.

In addition, it has to be mentioned that no much case studies existed within the sponsoring organization, due to ‘’information blockage’’ implied to the organization by the suppliers. However, even with this limited number of case studies, it was possible for a HW D&D effort estimation module to be created, which can offer valuable help to the estimator. In his attempt to perform a HW D&D effort estimation, the estimator has to be aware of the HW’s BOM; in case this is not disclosed by the supplier or when the estimator needs to judge the correctness of the supplied BOM, then the estimator has to apply expert knowledge either to derive one or to judge the one supplied.

10.2.4.3. Reuse and Integration Module

In the case of Reuse and Integration, neither the specifications nor any form of information existed. Therefore, an approach such as the one for the HW or SW metrics and module development could not be applied and an alternative path should be followed.

In the case of Reuse, reuse tables were developed. In these reuse tables, reuse percentages derived using expert knowledge. These reuse percentages represent expected amount of reuse, either for SW or HW, based on qualitative characteristics affecting the D&D effort of the embedded system. The logic is that a reuse matrix is developed for each and every one of the embedded systems included in the car. That means that the expert, depending on what characteristics his D&D environment has and for every item, he is able to retrieve the corresponding reuse table and choose the appropriate percentage of expected reuse to use it in his estimate. The difficulty on applying this approach is on data collection, since these percentages reside on the experts' minds and have to be collected and stored both for creating the reuse tables, as also for future use.

In the case of Integration, due to the vast variety of unpredictable side-effects that could occur on each of the integration phases (SW to SW, SW to HW, In-car integration), a quantitative approach was not possible to be applied; a qualitative approach was used instead, where a survey derived percentages for each of the integration categories. As in reuse, the expert is able to retrieve the appropriate integration percentage and use it in his estimate.

10.2.4.4. General Comments

Using data from past projects and the knowledge of experts, relationships were created in the form of equations or tables that could be associated to the elements of the cost estimate and predict the cost of the D&D effort. In the case of SW and HW, the system's specifications are transformed to metrics which are then used to predict the D&D effort. In the case of Reuse, a qualitative approach is followed, based on reuse percentages for each item and for several alternative D&D environments. In the case of Integration, integration percentages are used, derived by experts' knowledge.

All these come together to form the embedded system's D&D effort estimation framework. First, the HW D&D effort and the SW D&D effort are being estimated. Second, these two efforts are amended by the appropriate reuse percentage. At the end, the overall embedded system's D&D effort is calculated as the sum of the amended HW and SW efforts plus the Integration effort (which equals the 55% of the overall D&D effort: since SW and HW efforts are 45% of the overall D&D effort, then the 55% can be easily calculated).

Each of the framework's modules has been developed using case studies from different car domains (ie for SW, case studies from the infotainment domain were used, whereas for HW, case studies from Body domain) and, as it was later shown by the validation, each module in specific and the complete framework in general produced results very close to the actuals. Two points have to be mentioned in this point: (i) the framework could be expanded to cover the rest of the domains, if additional data collection and analysis is performed, and (ii) when the framework is implemented in another automotive organization, further validation is needed, for calibration reasons.

Considering the above, a structured approach for ES D&D effort estimation has been developed which could be easily transferred to other companies, provided that in the new company data collection, analysis and framework calibration would be performed in the new environment.

10.2.5. Framework Validation

The ES D&D effort estimation framework was created using case studies from the automotive industry. Each case study used in a module was independent of the rest of the case studies used in the remaining modules. The reason for this was that there was no complete case study to base the development of the entire framework.

Each module of the framework, as developed, was first validated by experts of the sponsoring organization and changes introduced when necessary. The framework's modules were tested for their accuracy with regard to the final cost or effort values produced. In all the cases, experts agreed that the results were realistic, that the modules (and therefore the framework) consisted an innovative approach on ES D&D effort estimation, and that this approach could significantly improve the quality of their estimates.

Once all the modules were developed and validated and after any amendments were introduced, the researcher validated the framework with another two OEMs, through workshops that were held in the OEMs premises. All the modules apart from SW were tested, both because of the limited availability of the experts from both OEMs, as well as because the SW module was extensively tested in previous steps.

During the validation of the framework, it was not possible for the experts to test its computerized version. This was done primarily because of the absence of a complete case study which could be used as an input for the excel-based implementation of the framework, and secondarily because of the time limitations the experts had.

With the two OEMs the framework was tested, module-by-module, for its completeness and its correctness. Validation was done in a module-by-module case (as in the sponsoring organization), because no complete case study existed within the two OEMs. In addition, this is the way these OEM work (they do not have a structured model/process which they use to run a case study through; they confront each part of the case study as a separate, isolated issue). The experts were presented each module, the rational and the way it works was explained to them and then they were presented the results it produces. The whole interviewing process was supported by a semi-structured questionnaire.

The experts commented in the usability, the effectiveness and the correctness of the framework. It was find easy to use and that it produces results close to actual values. Although recommendations were made that the framework could be further refined, the experts were very happy with its rational, the way it works and the results it produces. The fact that the framework was validated by two, additional to sponsoring organization OEMs, proves the validity of the model and assists on reducing bias regarding the testing process.

The researcher acknowledged these suggestions, however both the researcher and the experts came to the common agreement that spending more time and resources on making the framework more detailed would not be of a great benefit, since the framework was created for assessing the ES D&D cost on the specifications stage and for that, the results were already satisfactory.

10.3. Research contributions

The research has significantly contributed in better understanding the ES D&D domain as well as identifying the problems associated with the estimation of the ES D&D effort. It also introduced a novel approach for the estimation of this effort based on relationships developed between the system's specifications and the system's characteristics (ie functionality, number of components, cost, etc).

This framework allowed experts to bypass the ''supplier's blockage'' (the shortage of information because it is protected by supplier's IP rights) and adapt a structured approach for estimations in the specifications stage, as this became obvious through the workshops with the various OEMs. Therefore, the following points clearly identify the contribution to knowledge this research has made:

- The research, through its literature review, offered a wider visibility on the ES D&D domain, by identifying the issues that create problems during the D&D process and why these problems make the estimation process difficult. In addition, the importance of accurately estimating the ES D&D effort was also identified.
- The research, through its AS-IS study, verified the view obtained by literature regarding the importance of accurately estimating the ES D&D effort not only for the automotive but for other industries as well, whereas, at the same time, it identified the lack of information necessary to perform an ES D&D effort estimation.
- This research analysed factors that affect ES D&D effort estimation. Then, a step by step process has been developed to predict D&D effort for the HW, SW, Reuse and Integration parts of the ES.
- The research developed a systematic and consistent framework where experts can develop ES D&D effort estimates using relationships between the system's specifications, the system's characteristics and the system's D&D effort/cost, bypassing any information shortage.

10.4. Implementation Issues

The result of this research provided a framework with all the necessary elements to develop a cost estimate for automotive embedded systems. There were also instructions given on how and where these elements could be located. Finally, the framework was presented in the form of an excel implemented model, which is the complete working model that was provided for utilization to the sponsoring organization.

10.4.1. Technical Issues

The model should be flexible enough to allow progressive development, changing basic concept, adding further sensitivity and fine-tuning for greater accuracy of results. Therefore, Microsoft Excel was selected as a most suitable software tool to develop the user interface and the model itself.

This, provided an inexpensive and easy to use implementation as well as an inexpensive an easy way for testing the research and the framework too. Companies could use the model as a stand-alone SW or integrate it in their overall IT infrastructure. If the model is implemented as a stand-alone SW then a cost estimating database should be created storing past projects' data and information collected by experts across the organisation to support the operation of the framework and used in performing embedded systems cost estimation.

If the second implementation option is chosen (integrate it in the overall IT infrastructure), then links and interfaces to the programs the framework will be integrated with should be developed, so the expert would be capable of locating and retrieving the necessary for the cost estimating information and use them for performing a cost estimation with the model.

Integrating the model to the organisation's infrastructure would add many new features, such as copy functions, search functions, automatic update of cost rates, etc. This, however, in the case of the sponsoring organisation is a very difficult task since although the sponsoring organisation holds a very detailed cost database, this database only holds information about HW items; there are no information about SW, Reuse or integration. Therefore, for the correct operation of the model additional databases on SW and Reuse have to be created.

In both cases, external databases have to be created, to store SW projects and reuse percentages; therefore implementation issues have to be considered. Implementation issues are about the format and the structure of the database, since this will affect how the model connects to it and retrieves data from it.

10.4.2. Maintenance Issues

Maintenance issues need to be considered to ensure that both the framework and any stored data remain accessible and useable. If the framework remains in an Excel form or in a widely acceptable SW package (ie Microsoft Access) these issues are less pertinent. However, if the framework is implemented in an ad-hoc manner or in a proprietary SW, then the company has to be aware of any effects from changes in HW, operating systems or database versions and user request changes. Changes in any of these factors may affect the framework implementation and operation, and for that reason any change should be managed to ensure that both the model and the remaining infrastructure remain accessible and usable.

10.4.3. Financial Issues

The model has been designed, developed and implemented in Excel, which reduces the financial burden. However, the cost of the underlying database (for SW projects and for the reuse data) has to be considered, since this choice will affect the implementation and the development cost. Increased cost will also occur if the company chooses to implement the model in a different (than to

Excel) more complicated way. The customisation of the model should also be considered.

To compensate for the implementation and the development costs, both the long and short term benefits from the SW tool (improved rigor, consistency, improved negotiations with the customers, estimates reuse, improved confidence with the estimates, etc) should be taken into consideration and be promoted both to the senior management and the users.

10.4.4. Cultural Issues

Cultural issues are usually a big contributor to the difficulty organisations face to change. The framework found great appeal within the automotive industry. Indeed, all three automotive OEMs participating in this research plan to further adopt the model, calibrate it and use it for more commodities. The cost estimating framework also promotes the demise of cultural barriers, as during its implementation and operation it allows experts from various areas to work together to collect data, develop and calibrate the model and develop the cost estimate.

What is significantly affected by the adoption and the operation of this framework is the OEM-Supplier relationship. Throughout this research, it has been identified that for OEMs the major obstacle on performing a cost estimate in the specifications stage is the absence of data, data that are protected by Suppliers' IPR and are not disclosed to the OEM. This ''information blockage'' was the reason for initiating this research and for creating the embedded systems D&D effort (and therefore cost) estimation framework.

After the creation of the framework, OEMs have a ''tool'' that can help them arrive in an estimate for the commodity to be developed. Data, knowledge and information will be shared and OEMs would be much more confident when embarking in negotiations with the suppliers. As a result, the suppliers would not longer be able to withhold information or to deliver a high price without justification.

10.5. Limitations

- This research was heavily dependent on the resources and support from the industry. Much of the work between both the sponsoring organization as well as the rest of the participating organisations is of a confidential nature for the organisations. Therefore, sensitive for the organisations comments have not been recorded on tape, which means that in some cases the researcher had to rely only on his notes and/or memory, incurring the danger of increasing the researcher's bias. In order to provide validity and reliability and minimize the bias coming from his prolonged involvement within the participating organisations, the researcher used multiple sources of information, maintained a chain of evidence and recorded interviews (where this was possible). This does not completely eliminate the researcher's bias, but does provide a means for other researchers to examine how the results were obtained.

- The people that participated in the workshops or interviews on each module development as well as in the validation of the framework were people chosen by the participating organisations because of their expert knowledge. The selection of senior people was necessary because, using their experience, they could offer the best feedback on the research. However, experts were not always available; this limited both the interview chances as well as the duration of each interview (or workshop).
- The number and type of case studies was predetermined, as they were not many case studies and/or corresponding combinations available (ie. in the HW module where case studies with their corresponding BOM were requested). Therefore, individual case studies were used in the development of each of the framework's modules, since these were the case studies the organisations had available for each ES domain. For example: for the embedded SW estimation based on UML Use Cases specifications module development, use cases coming from the car's Information and Entertainment (Infotainment) domain were used, whereas for the embedded SW estimation based on Statecharts specifications module development, use cases coming from the car's Body domain were used. For that reason the validation of the framework also had to be performed in a module-by-module basis only.
- The validation of the $IDEF_0$ models was by the experts that helped develop the models. This could cause bias since the models represent their view of the process being modelled. A better method would have been to use other domain experts not involved during the development of the models.
- The choice of using Excel to implement the framework in a computerised form was due to the author's limitations of higher level programming skills. The computerised framework might have been better implemented ie using a programming language or a commercial SW package, or as an add-in feature to other programmes ie such as Access.

- In some modules of the framework (ie HW), 'drop-down' lists were used to control the data input. However, much of the data required within the framework depend on the judgement of the expert (ie Reuse). Selecting assumptions, exclusions, and rationale from predetermined lists would help to minimise errors, and reduce the time required to input data.
- The author had to validate the framework qualitatively, since neither the time nor the absence of complete case studies permitted its validation otherwise. Therefore, qualitative validation was the best way to gather feedback. In addition, qualitative validation offered a deeper understanding of the experts' views. Although all experts stated that the framework could be very useful to them, the author would have liked to see the application of the framework on one (or more) complete case studies.
- The framework was developed and tested only within the automotive industry. The applicability of the framework in other industries has to be examined.

10.6. Future Research

The importance of ES D&D cost estimating has been stated within this thesis. Evidence has been derived that accurate estimation of the D&D effort is an important issue for the automotive industry. This acquires special importance in the specifications stage of the development process, where little –or no- information is available to the expert.

During this research, the author identified a lack of literature regarding the ES D&D effort (and therefore cost) estimation modeling. It seems that there is a tendency in the ES cost estimating domain to be concentrated in manufacturing. There is the need for more detailed analysis of the ES D&D domain, in order to improve its understanding.

The researcher created an ES D&D effort estimation framework for the specifications stage and for the automotive industry. This framework is based on various data, which the researcher explained –during the framework's development- why they are needed and where they could be obtained from. It is suggested by the researcher the actual gathering of these data as a step forward in the framework development. Data collected by the sponsoring organization experts would enhance framework's accuracy; data gathered by experts in other industries would enhance frameworks generalisability and portability.

The framework created was found very useful by the automotive industry. In order to further enhance its ease of use, the researcher implemented the framework in an automated form using Microsoft Excel. This automated approach could be further enhanced by being able to capture, store and incorporate any of the experts' assumptions made and used during the estimates. This way, the assumptions could be retrieved and reused by the experts.

10.7. Conclusions

In paragraph 3.1, the following objectives were set:

- To identify state of the art research in electronic parts cost estimation and in particular on estimating their D&D cost within the manufacturing industry.
- To assess the challenges organisations face when estimating the D&D cost of an electronic item within the automotive industry.
- To link the specifications of an electronic item with the cost estimating process.
- To develop a framework to structure and formalise the cost estimating process within an automotive OEM for ES D&D. The framework will include HW, SW, Reuse issues and their Integration and Testing.

The research has achieved all the objectives mentioned in Chapter 3. The main conclusions from the research are presented bellow:

1. The importance of the ES D&D cost estimation for the automotive industry has been identified.
2. It is observed that there is a lack of research regarding ES D&D cost estimation.
3. It is observed that this problem is of the greatest importance in the specifications stage. This is due to lack of information with automotive OEM at the specification stage.
4. The research has identified metrics, data and information required to avoid this information shortage.
5. The research has demonstrated that a complexity based framework can be developed to predict hardware D&D effort. This is a formal approach to the cost estimating.
6. The study has addressed the cost estimation for Software using both Use Case and Statechart specifications. A combination Use Case points and expert judgment can predict the effort required for the software D&D. The research has linked ES specification directly with the software D&D effort.

7. The thesis investigates the factors that affect integration and reuse effort involved in ES D&D within the automotive industry. An expert judgment based framework is presented to predict the integration and reuse effort.
8. The framework was validated by 3 OEMs and through this validation it was observed that this framework presents a formal and structured approach to the cost estimating at the early stage ES development.
9. The research has improved understanding about ES D&D effort required in the automotive sector.

References

1. AACE, ''AACE international recommended practices and standards'', AACE International, CDROM, ITEM NUMBER: 4060-05.

2. Anda, B., Anhelvik, E., and Ribu, K., ''Improving estimation practices by applying Use Vase Models''. Lecture Notes in Computer Science, vol 2559, pp 383-397, 2002

3. Anda, B., Dreiem, H., Sjoberg, D. I. K., and Jorgensen, M., ,'Estimating software development effort based on Use Cases – experiences from Industry'', Proceedings of the 4th International Conference on the UML, Lecture Notes in Computer Science , Vol. 2185, Gogolla, M., and Kobryn, C. (Eds.), pp. 487-502, 2001

4. Arnold, M. and Pedross, P., ''Software size measurement and productivity rating in a large scale software development department'', Proceedings of the 20th International Conference on Software Engineering, IEEE Computer Society, pp. 490-494, 1998

5. Automotive UML: www.automotive-uml.org

6. Axelsson, J., 'Hardware/Software Partitioning of Real-Time Systems'. In IEE Colloquium on Partitioning in Hardware-Software codesigns, London, pp. 1-8, February 1995.

7. Axelsson, J., 'Analysis and Synthesis of Heterogeneous Real-Time Systems'. PhD Thesis, Linköping University, Sweden, 1997a.

8. Axelsson, J., 'A Hardware/Software Codesign Approach to System-Level Design of Real-Time Applications'. Swedish National Real-Time Systems conference (SNART 97), Lund, Sweden, August 21-22, 1997b.

9. Axelsson, J., 'Towards System-Level Analysis and Synthesis of Distributed Real-Time Systems'. 5th International Conference on Information Systems Analysis and Synthesis Proceedings, Vol. 5, pp. 40-46, Orlando, July 31-August 4, 1999.

10. Axelsson, J., ‘Methods and Tools for Systems Engineering of Automotive Electronic Architectures’. Proceedings of the Design Automation and Test in Europe Conference, CR-ROM version, Munich, March 13-16, 2001

11. Axelsson, J., ‘Model Based Systems Engineering using a continuous-time extension of the Unified Modeling Language (UML)’. Journal of Systems Engineering, vol.5, issue 3, pp. 165-179, 2002a.

12. Axelsson, J., ‘’Research challenges in automotive real-time systems’’, ARTIST workshop, Paris, 23rd of April, 2002b. Downloaded from http://www.artist-embedded.org/PastEvents/Kickoffs/index.html, accessed 10 of June, 2004

13. Balarin, F., Hsieh, H., Watanabe, Y., Lavagno, L.; Passerone, C., and Sangiovanni-Vincentelli, A., ‘Metropolis: an integrated electronic system design environment’. Computer magazine, vol. 36, pp. 45-52, April 2003

14. Balarin, F., Lavagno, L., Passerone, C., and Watanabe, Y., ‘Processes, Interfaces and platforms. Embedded Software Modeling in Metropolis’. Embedded Software Conference (EMSOFT) Proceedings, pp.407-421, 2002

15. Balarin, K., Giusto, P., Jureska, A., Passerone, C., Sentovich, E., Tabbara, B., Chiodo, M., Hsieh, H., Lavagno, L., Sangiovani – Vincentelli, A., and Suzuki, K., ‘Hardware-Software Co-Design of embedded systems, the Polis approach’. Kluwer Academic Publishers, April, 1997

16. Balboni, A., Fornaciari, W., and Sciuto, D., ‘Co-synthesis and Co-simulation of control dominated embedded systems’. Design Automation for embedded systems, vol. 1, no. 3, pp. 257-289, 1996

17. Ball, R. S., 1996. ‘Embedded microprocessor systems: real world design’, Newnes Publishing (an imprint for Butterworth-Heinemann)

18. Bashir, H. A. and Thomson, V., ‘’Metrics for design projects: a review’’, Design studies, vol. 20, no. 3, pp. 263-277, May 1999

19. Beeck, M. v. d., Braun, P., Rappl, M., and Schroder, C., '‛Automotive UML: A (Meta) Model Based approach for Systems Development'', UML for Real, Kluwer Academic Publishers, 2003

20. Bell, D., '‛UML Basics: an introduction to the Unified Modeling Language'', The Rational Edge, June 2003

21. Berger, A.S., 'Embedded Systems Design: An Introduction to Processes, Tools and Techniques'. CMP Books, 2002, 1st edition

22. Berry, G., 'Real time programming: Special purpose of general purpose languages'. In Information Processing, North Holland-Elsevier Science Publishers, 1989

23. Bloch, C., and Ranganathan, R., '‛Process Based Cost Modeling'', IEEE Transactions on Components, Hybrids, and Manufacturing Technology, vol15, no3, pp. 288-294, June 1992

24. Boehm, B. W., 'Software Engineering Economics', Prentice Hall, 1981.

25. Boehm, B.W. and Harrowith, E., 'Software Cost Estimation with COCOMO II'. Prentice Hall, August 2000

26. Bouyssounouse, B., and Sifakis, J., '‛Current design practice and needs in selected industrial sectors'', Lecture Notes in Computer Science (LNCS) no.3436: '‛Embedded systems design: The ARTIST Roadmap for research and development'', pp. 15-38, Springer-Verlag GMBH, 2005

27. Butts, K., '‛Modeling to Manage Electronic Complexity'', ARTIST International Collaboration Day on Embedded Software and Systems, 6th of October, Grenoble, France, 2002. Downloaded from http://www.artist-embedded.org/PastEvents/ICD_2002/index.html, accessed 10 of June, 2004

28. Calvez, J.P., 'Embedded real-time systems: a specification and design methodology'. John Willey, 1993

29. Chan, F., Spiller, M., and Newton, R., 'WELD – an environment for web-based electronic design'. Proceedings of the Design Automation Conference, pp. 146-151, 15-19 of June, San Francisco, 1998

30. Chen, R., Sgroi, M., Lavagno, L., Martin, G., Sangiovanni-Vincentelli, A., and Rabaey, J., ''UML and Platform-Based design'', in: ''UML for real'' Kluwer Academic Publishers, 2003

31. Chiodo, M., Giusto, P., Jureska, A., Hsieh, H., Sangiovani – Vincentelli, A., and Lavagno, L., 'Hardware-Software Co-Design of embedded systems', IEEE Micro, vol. 14, no. 4, pp. 26-36, August 1994

32. Chou, p., Ortega, R., and Borriello, G., 'the Chinnok Hardware/Software Co-synthesis system'. International symposium on System Synthesis, Cannes, France, September 1995

33. Cornu, X., and Kung, A., 'AJACS: Applying Java to Automotive Control Systems'. AJACS project concluding paper. Downloaded from http://www.ajacs.org/documents/AJACS-concluding-paper-public-version.pdf, accessed 20/09/2003

34. Cronenweath, S., ''The route to higher quality automotive software: An interview with Andreas Eppinger, IBM Germany, and Kevin Mixer, AMR Research'', The Rational Edge, June 2004 issue.

35. DACS (Data and Analysis Center for SW: a Department of Defense (DoD) Information Analysis Center), ''Embedded Software Maintenance'', Technical Report, February 2003, downloaded from http://www.dacs.dtic.mil/techs/embedded, accessed 20 of December 2003

36. Davis, J. II., Hylands, C., Janneck, J., Lee, E., Liu, J., Liu, X., Neuendorffer, S., Sachs, S., Stewart, M., Vissers, K., Whitaker, P., and Xiong, Y., 'Overview of the Ptolemy project'. Technical Memorandum UCB/ERL M01/11, University of California at Berkeley, March 2001

37. Debardelaben, A. J., Madiseti, J. A., and Gadient, J. A., 1997. ‘Incorporating cost modeling in embedded system design’, IEEE Design and test of computers, pp 24-35, July-September 1997.

38. Department of Trade and Industry (DTI), ‘’Design reuse and intellectual property: an introduction’’, DTI Electronics Design Publications, March 2004a

39. Department of Trade and Industry (DTI), ‘’A manager’s guide’’, DTI Electronics Design Publications, March 2004b

40. Department of Trade and Industry (DTI), ‘’Best Practice’’, DTI Electronics Design Publications, March 2004c

41. Di Natale, M., and Domenici, A., ‘The DASE Tool: An Environment for the Development of Real-Time Applications’. Proceedings of the International Workshop on Distributed and Parallel Embedded Systems (DIPES '98), pp. 169-180, 5-6 of October 1998

42. DIN 40050/IEC 529 standard, ‘’www.din.org’’, accessed 10th of August, 2005, 12:30am

43. DoD Parametric cost estimating handbook 2nd edition, 1999

44. DoD Parametric cost estimating handbook 3rd edition, 2003

45. Dömer, R., Gajski, D., and Zhu, J., ‘Specification and Design of Embedded Systems’. IT+TI magazine, Oldenbourg Verlag, Munich, Germany, No. 3, June 1998.

46. Douglass, B. P., ‘Real Time UML: Developing efficient objects for embedded systems’. 2nd edition, Addison – Wesley, 1999

47. Drusinsky, D., and Harel, D., ‘Using Statecharts for hardware description and synthesis’. IEEE Transactions on Computer Aided Design, vol. 8, no. 7, pp. 798-807, 1989

48. Duverlie, P. and Castelain, J. M., "Cost estimation during design step: Parametric method versus case based reasoning method", International Journal of Advanced Manufacturing Technology, v15, pp. 895-906, 1999

49. Department of Trade and Industry (DTI), ''Design reuse and intellectual property: an introduction'', DTI Electronics Design Publications, March 2004a

50. Edmonds, B., "Syntactic Measures of Complexity", Doctoral Thesis, University of Manchester, Manchester, UK, 1999.

51. Edwards, C., ''European auto makers seek control over subsystems'', www.eedesign.com/story/OEG20010316S0058, accessed 1st of December, 2004

52. Edwards, S., Lavagno, L., Lee, E. A., and Sangiovanni-Vincentelli, A., 1997. 'Design of embedded systems: formal models, validation and synthesis', Proceedings of the IEEE, vol85, no3, pp. 366-390, March 1997

53. Eles, P., 'Codesign of embedded systems: where are we now?'. Embedded Systems Laboratory (ESLAB), Department of Computer and Information Science, Linköping University, Sweden, http://www.ida.liu.se/~petel/, assessed 8th of July, 2003

54. Engels, D. W., and Devadas, S., ''A new approach to solving the Hardware-Software partitioning problem in embedded system design'', Proceedings of the 13th Symposium on Integrated Circuits and Systems Design, pp. 275-280, 18-24 of September, 2000

55. Erdos, R., ''Human factors: Test Pilot's Viewpoint'', Aviation Today, 21st of March, 2005

56. Ernst, R., et al, 'The Cosyma Environment for Hardware/Software Cosynthesis of small embedded systems'. Microprocessors and Microsystems, vol.20, no.3, pp. 159-166, May 1996

57. Ernst, R., 'Codesign of embedded systems: status and trends'. IEEE Design and Test of Computers, vol. 15, no. 2, pp. 45-54, April-June 1998

58. Farineau, T. et al, 'Use of parametric models in an economic evaluation step during the design phase', In: International journal of advanced manufacturing technology, 2001, v17, pp.79-86.

59. Feiler, H. P., Lewis, B., and Vestal, S., 'Improving predictability in embedded real-time systems', CMU/SEI – 2000-SR-011 Special report, December 2000

60. Ferri, R. N., Pratiwadi, R. N., Rivera, L. M., Shakir, M., Snyder, J. J., Thomas, D. W., Yin Farn, C., Fowler, G. S., Krishnamurthy, B., and Kiem-Phong, V., ''Software Reuse Metrics for an Industrial Project''. Proceedings of the 4th International Software Metrics Symposium, pp. 165-173, 5-7 of November, 1997.

61. Foulger, R. J., 1982. 'Programming embedded microprocessors: a high-level language solution', NCC Publications.

62. Frakes, W., and Terry, C., ''Reuse Level Metrics'', Proceedings of the 3rd International Conference on Software Reuse: Advances in Software Reusability, pp. 139-148, 1-4 of November, 1994

63. Frakes, W., and Terry, C., ''Software reuse and reusability metrics and models'', TR-95-07 Technical Report, Computer Science, Virginia Polytechnic Institute and State University, 1996a

64. Frakes, W., and Terry, C., ''Software Reuse: Metrics and Models'', ACM Computing surveys, vol28, no2, pp. 234-247, June 1996b

65. Frederic, B. C., and Jago, W. H., ''A model for cost estimating modern electronic boxes'', Proceedings of the IEEE, vol42, pp. 6.2.1-6.2.6, 26-30 of October, 1997

66. Fowler and Scott, K., ''UML Distilled: Second edition. A brief guide to the standard object modeling language'', 2nd edition, Addison-Wesley Logman Inc, 2002

67. Gajski, D., Zhu, J., and Dömer, R., Essential issues in Codesign', Technical Report, ICS-97-26, Dept. of Information and Computer Science, University of California, Irvine, June 1997

68. Gajski, D., Narayan, S., Vahid, F., and Gong, J., 'Specification and Design of embedded systems'. Prentice-Hall, 1994

69. Gajski, D., Aggarwal, G., Chang, E., Dömer, R., Ishii, T., Kleinsmith, J., and Zhu, J., 'Methodology for design of embedded systems', Technical Report, UCI-ICS-98-07, Dept. of Information and Computer Science, University of California, Irvine, March 1998

70. Gajski, R.D., Dömer, R., Zhu, J., 'IP-centric Methodology and Design with the SpecC Language' Chapter 10 in System-Level Synthesis, Chapter 10 in ''System-Level Synthesis'', Proceedings of the NATO ASI on System Level Sythesis for Electronic Design, Il Ciocco, Lucca, Italy, Aug. 1998, Ahmed, A. J., and Jerraya, J. P. M., Kluwer Academic Publishers, May 1999.

71. Geiger, T.S. et al, 'Automated design-to-cost: integrating costing into the design decision', In: Computer-aided design, 1996, v.28, pp.423-438.

72. Graaf, B., Lormans, M., and Toetenel, H., ''Embedded Software Engineering: The state of the practice'', IEEE Software, vol. 20, no. 6, pp. 61-69, November/December 2003

73. Graves, R. J., Agrawal, A., and Haberle, K., ''Estimating tools to support multipath agility in electronics manufacturing'', IEEE Transactions on Components, Packaging and Manufacturing Technology, part C, vol19, no1, pp. 48-56, January 1998

74. Grehan, A., Moote, J., and Cyliax, E., 'Real time Programming', Addison – Wesley, 1998

75. Greves, D., and Schreiber, B., 'The ESA initiative for software productivity, benchmarking and effort estimation', ESA Bulletin no87, August 1997

76. Gupta, R., ‘Introduction to embedded systems’, www1.ics.uci.edu/~rgupta/ics212/w2002/intro.pdf, accessed 10th of August 2003

77. Harel, D., ‘STATEMATE: a working environment for the development of of complex reactive systems’. IEEE transactions in Software Engineering, vol. 16, no.4, pp. 403-414, April 1990

78. Heath, S., ‘Embedded Systems Design’, Newnes publishing (an imprint of Elsevier Science), 2nd Edition, 2003

79. Heemstra, F. J., 1992. ‘Software cost estimation’, Information and Software technology, vol34, no10, pp 627-639, October 1992.

80. Heikkinen, M., ‘A concurrent engineering process for embedded systems development’. VTT Publications, Espoo, Finland, 1997

81. Henzinger, T.A., Horowitz, B., and Kirch, C.M., ‘Giotto: a time-triggered language for embedded programming’. Embedded Software Conference (EMSOFT) Proceedings, pp.166-184, 2001

82. Hsieh, H., Balarin, F., Lavagno, L., and Sangiovanni-Vincentelli, A., ‘Efficient methods for embedded system design space exploration’. 2000 Design Automation Conference Proceedings, pp. 348-360, 5-9 of June, 2000

83. Hughes, R. T. (1996), “Expert judgement as an estimating method. Information and Software Technology”, v38, pp: 67-75.

84. Humphrey, S. W., ‘Introduction to Software Process Improvement’. SEI Techical report CMU/SEI-92-TR-007, June 1992

85. ICEC, ‘About Standardisation’, http://www.icoste.org, (Accessed June 2004)

86. Innes, J. et al, *‘Activity costing for engineers’*, John Wiley, New York, 1994.

87. Ismail, T., Abid, M., and Jerraya, A., 'Cosmos: a codesign approach for communicating systems'. Proceedings of the Third International Workshop on Hardware/Software Codesign, IEEE Computer Society Press, pp. 17-24, 1994

88. Jacome, M. F., and Lapinskii, V., ''NREC: Risk Assessment and Planning of Complex designs'', IEEE Design and Test of Computers, pp. 42-49, January-March 1997 issue.

89. Jeffery, R., and Walkerden, F., 'Analogy, regression and other methods for estimating effort and software quality attributes', In: Project Control for software quality, Kusters, Cowderoy, Heemstra and van Veenendaal (eds.), Shaker Publishing, 1999

90. Jerraya, A. A., and Wolf, W., ''Hardware/Software Interface Codesign for embedded systems'', IEEE Computer, vol. 38, no. 2, pp. 63-69, February 2005

91. Jersak, M., Richter, K., Ernst, R., Braam, J. C., Jiang, Z. Y., and Wolf, F., ''Formal Methods for Integration of Automotive Software'', Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 45-50, 2003

92. Josifovska, S., 'Stalemate', IEE Review, April 2003 issue.

93. Kalavade, A., and Lee, E., 'A Hardware-Software Co-Design methodology for DSP applications'. IEEE Design and Test of Computers, vol. 10, no. 3, pp. 16-28, September 1993

94. Kang, K. C., Cohen, S., Holibaugh, R., Perry, J., and Peterson, A. S., ''A reuse based software development methodology', CMU/SEI–92-SR-4 Special report, January 1992

95. Karsai, G., Sztipanovits, J., Ledeczi, A., and Bapty, T., 'Model integrated development of embedded software', Proceedings of the IEEE, vol91, no1, pp. 145-164, January 2003

96. Keremer, F. C., ‘An empirical validation of software cost estimation models’, Communications of the ACM, vol30, no5, pp 416-429, May 1987.

97. Klein, R.A., ‘Miami: A Hardware-Software Co-simulation environment’. Proceedings of the Seventh International Workshop on Rapid System Prototyping, IEEE Computer Society Press, pp. 173 – 177, 19-21 of June, 1996

98. Koopman, P., ‘Embedded systems design issues: the rest of the story’. Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 310 – 317, 7-9 of October, 1996

99. Kopetz, H., ‘Real time systems: design principles for distributed embedded applications’. Kluwer Academic Publishers, April 1997

100. Jusumoto, S., Matukawa, F., and Inoue, K., ‘’Estimating effort by Use Case Points: Method, Tool and Case Study’’, Proceedings of the 10th International Symposium on Software Metrics (METRICS), pp. 292 – 299, 14-16 of September, 2004

101. Lavagno, L., and Sangiovanni-Vincentelli, A., ‘System-level design models and implementation techniques’. In IEEE International Conference on Application of Concurrency to System Design, pp. 24 – 32, March 1998

102. Lavagno, L., Sangiovani – Vincentelli, A., and Sentovich, E., ‘Models of computation for embedded systems design’. NATO ASI on System Level Sythesis for Electronic Design, Il Ciocco, Lucca, Italy, Aug. 1998, Ahmed, A. J., and Jerraya, J. P. M., Kluwer Academic Publishers, May 1999.

103. Lawrence, P. D., and Mauch, K. ‘Real-Time Microcomputer System Design: an Introduction’, McGraw-Hill, 1998.

104. Lederer, A. L., and Prasad, J., ‘Information systems software cost estimating: a current assessment’, Journal of Information Technology, vol 8, pp. 22-33, 1993.

105. Lee, A. E., ‘System level design methodology for embedded signal processors’. Ptolemy project final report, University of California at Berkeley, September 1997

106. Lee, A. E., 2000. ‘What is ahead for embedded software’, Computer magazine, IEEE Computer Society Publications, v33, issue 9, pp. 18-26, Sept. 2000

107. Lee, A. E., ‘Embedded software’, Advances in Computers, Academic Press, vol56, London, 2002

108. Lee, E., and Sangiovanni-Vincentelli, A., ‘A framework for comparing models of computation’. IEEE transactions on computer aided design of integrated circuits and systems, vol17, no12, pp. 1217-1229, December 1998

109. Liao, S., Tjiang, S., and Gupta, R., ‘An efficient implementation of reactivity for modelling hardware in the Scenic design environment’. Proceedings of the 34th Design Automation Conference, pp. 70-75, Anaheim, California, USA, 1997

110. Liu, X., Liu, J., Eker, J., and Lee, E., ‘Hetereogeneous Modelling and Design of Control Systems’. In Software-Enabled Control: Information technology for Dynamical Systems, Tariq Samad and Gary Balas (eds), Wiley-IEEE Press, April 2003

111. Lloyd, S., “Programming the Universe”, Knopf, 2006

112. Londeix, B., ‘’Software: An engineering discipline?’’, IEE Review, September 2003 issue.

113. Lother, M., Dumke, R. R., Bohm, T., Herweg, H., and Reiss, W., ‘’Applicability of COSMIC Full Function Points for BOSCH specifications’’, Proceedings of the 13th International Workshop on Software Measurement, pp. 204-217, Montreal, 2003

114. Maciaszek, L. A., ‘’Requirements analysis and system design: Developing information systems with UML’’, Pearson Education Limited, 2001

115. Madsen, J., Grode, J., and Knudsen, P., 'Hardware-Software partitioning using the LYCOS system'. In: Hardware/Software Co-Design: Principles and Practice, edited by J. Staunstrup and W.Wolf, Kluwer Academic publishers, 1997

116. Martin, G., Lavagno, L., and Louis-Guerin, 'Embedded UML: a merger of real time UML and co-design', CODES 2001, Denmark, April 2001

117. McKinsey&Company, 'Automotive Software: A battle for value', Study, 2002

118. Micheli, G. D., and Gupta, R. K., ''Hardware/Software codesign'', Proceedings of the IEEE, vol85, no3, pp. 349-365, March 1997

119. Morisio, M., Tully, C., and Ezran, M., ''Diversity in Reuse Processes'', IEEE Software, vol. 17, no.4, pp. 56-63, July/August 2000

120. Mukhopadhay, T., Vicinanza, S., and Prietula, M. J., 'Estimating the feasibility of a case base reasoning model for software effort estimation', MIS Quarterly, December 1992 issue.

121. Narayan, S., Vahid, F., and Gajski, D., 'System Specification and synthesis with the SpecCharts language'. Proceedings of the 1991 IEEE International Conference on Computer Aided Design, pp. 266 – 269, 11-14 of November, 1991

122. NASA Parametric cost estimating handbook 1st edition, 1995

123. Oestereich, B., ''Developing software with UML: Object Oriented analysis and design in practice'', 2nd edition, Pearson Education Limited, 2002

124. Olukotun, K., Heinrich, M., and Ofelt, D., ''Digital System Simulation: Methodologies and Examples'', Proceedings of the 35th Design Automation Conference, pp. 658 – 663, 15-19 of June, 1998, San Francisco, California, United States

125. Ortega, R. B., and Borrielo, G., ‘’Communication synthesis for embedded systems with global considerations’’, Proceedings of the 5th International Workshop on HW/SW Codesign, pp. 69-73, 24-26 of March, 1997.

126. Paulin, P. G., Liem, C., Cornero, M., Nacabal, F, and Goosens, G, 1996. ‘Trends in embedded systems technology: an industrial perspective’, Hardware-Software co-design, G. De Micheli and M. Sami (eds), Kluwer Publications, 1996

127. Paulin, P. G., Liem, C., Cornero, M., Nacabal, F, and Goosens, G, 1997. ‘Embedded software in real time systems: application and architecture trends’, Proceedings of the IEEE, vol85, no3, pp 419-435, March 1997.

128. Paulin, P. G., Liem, C., Cornero, M., Nacabal, F, and Goosens, G, 1997. ‘Embedded software in real time systems: application and architecture trends’, Proceedings of the IEEE, vol85, no3, pp 419-435, March 1997.

129. Peters, J. F. and Pedrycz, W., ‘’Software Engineering: an engineering approach’’, John Willey and sons Inc, 2000

130. Pimentel, A.D., Hertzberger, L.O., Lieverse, P., van der Wolf, P., and Deprettere, E.F., ’Exploring embedded systems architectures with Artemis’. Computer Magazine, November 2001 issue.

131. Preston, J. D., ‘’Perspectives on tool integration for embedded systems’’, ARTIST International Collaboration Day, 6th of October, 2002. Downloaded from http://www.artist-embedded.org/PastEvents/ICD_2002/index.html, accessed 10th of July, 2004

132. Probasco, L., ‘’Dear Dr. Use Case: What about Function Points and Use Cases?’’, The Rational Edge, August 2002

133. Putnam H. L. and Myers W., ‘What we have learned’. CrossTalk: The Journal of Defense Software Engineering, June 2000 Issue.

134. Ragan, D., Sandborn, P., and Stoaks, P., ‘’A detailed cost model for concurrent use with software/hardware co-design’’, Proceedings of the

39th IEEE/ACM Design Automation Conference, pp. 269-274, New Orleans, June 2002

135. Rumbaugh, J., Jacobson, I., and Booch, G., ‵′The Unified Modelling Language Reference Manual′′, 2nd edition, Pearson Education Inc, 2004

136. Rompaey, K., Verkest, D., Bolsens, I., and De Man, H., ‵CoWare codesign of DSP systems′. NATO ASI on System Level Sythesis for Electronic Design, Il Ciocco, Lucca, Italy, Aug. 1998, Ahmed, A. J., and Jerraya, J. P. M., Kluwer Academic Publishers, May 1999.

137. Rompaey, K., Verkest, D., Bolsens, I., and De Man, H., ‵CoWare, a design environment for hetereogeneous Hardware-Software systems′. European Design Automation Conference Proceedings, pp. 252-257, September 1996

138. Roy R., Bendall D., Taylor J.P., Jones P., Madariaga A. P., Crossland J., Hamel J. & Taylor I. M. ‵Identifying and Capturing the Qualitative Cost Drivers within a Concurrent Engineering Environment′, Advances in Concurrent Engineering, Chawdhry, P.K., Ghodous, P., Vandorpe, D. (Eds), Technomic Publishing Co. Inc., Pennsylvania (USA), pp. 39-50, 1999a.

139. Roy, R., Bendall D., Taylor J.P., Jones P., Madariaga A. P., Crossland J., Hamel J. & Taylor I. M. ‵Development of Airframe Engineering CERs for Military Aerostructures′, Second World Manufacturing Congress (WMC'99), Durham (UK), September 27- 30, pp. 838-844, 1999b.

140. Roy, R, Chim, D., Khan, F, Nwoyeocha, I. N. C, Robson, D. P., and Swainon, A.A.G., 2000. ‵Software cost estimating: research and practice′, IASTED international conference on software cost engineering and applications, Las Vegas, Nevada, USA, pp 813-818, November 2000

141. Ruhe, M., ‵Comparative Study of project effort estimation methods by using public domain, multi-organisational and organisation specific project data′, Technical Report, Center for advanced empirical software research (CEASAR), School of Information Systems, University of new South Wales, Sydney, August 1999

142. Rush, C., and Roy, R., 'Analysis of cost estimating processes used within a concurrent engineering environment throughout a product life cycle'. 7th ISPE International Conference on Concurrent Engineering: Research and Applications, pp. 58-67, Lyon, France, July 17th - 20th, 2000.

143. Rush, C., "Formalisation and reuse of cost engineering knowledge", PhD thesis, Cranfield University, United Kingdom, 2002.

144. Sangiovanni-Vincentelli, A., 'System Level Design with Embedded Platforms', Proceedings of the 37th ACM-IEEE Design Automation Conference, Los Angeles, California, United States, 5-9 of June, 2000

145. Sangiovanni-Vincentelli, A., ''Embedded system design for the automotive supply chain'', ARTIST International Collaboration Day, 6th of October, 2002. Downloaded from http://www.artist-embedded.org/PastEvents/ICD_2002/index.html, accessed 10th of July, 2004.

146. Sangiovanni-Vincentelli, A., 'Electronic system design in the automobile industry'', IEEE Micro, vol. 23, no. 3, pp. 8-18, May-June 2003

147. Sangiovanni-Vincentelli, A., 'Integrated electronics in the car and the design chain evolution or revolution?'', Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 532-534, 7-11 of March, Messe, Munich, Germany, 2005

148. Sangiovanni-Vincentelli, A., and Martin, G., 'Platform based design and software design methodology for embedded systems'', IEEE Design and Test of Computers, vol. 18, no. 6, pp. 23-33, November-December 2001

149. Sastry, S. S., Sztipanovits, J., Bajcsy, R. and Gill, H., 2003. 'Scanning the issue: special issue on modeling and design of embedded software', Proceedings of the IEEE, vol91, no1, pp3-10, January 2003

150. Scheffler, M., Ammann, D., Thiel, A., Habiger, C., and Troster, G., ''Modeling and Optimising the costs of electronic systems'', IEEE Design and Test of Computers, vol. 15, no. 3, pp. 20-36, July - September 1998

151. Schmietendoff, A., Dimitrov, E., Dumke, R., Foltin, E, and Wipprecht, M., ''Conception and experience of metrics based software reuse in practice'', International workshop on software measurement (IWSM), pp. 178-189, 8-10 of September, Lac Superieur, Quebec, Kanada, 1999

152. Sciuto, D., ''Guest editor's introduction: Design tools for embedded systems'', IEEE Design and Test of Computers, vol. 17, no. 2, pp. 11-13, April-June 2000

153. Selin, D., 'MESC Modelling Guideline: Model Based Electrical System Concept'. Internal Technical Report, Volvo Car Corporation, August 2002

154. Selic, B., and Rumbaugh, J., ''Using UML for modeling complex real-time systems'', Rational Object Time, March 1998

155. Sgroi, M., Lavagno, L., and Sangiovani – Vincentelli, A., 'Formal models for embedded systems design'. IEEE Design and Test of Computers, vol. 17, no. 2, pp. 14-27, April-June 2000 issue

156. Sgroi, M., Lavagno, L., Watanabe, Y., and Sangiovani-Vincentelli, A., 1999. 'Synthesis of Embedded software using free-choice petri-nets', Proceedings of the 36th Design Automation Conference, pp. 805-810, 21-25 of June, 1999

157. Shepperd, M., and Schofield, C., 'Estimating software project effort using analogies', IEEE Transactions on Software engineering, vol23, no11, pp. 736-743, November 1997

158. Shepperd, M., Schofield, C., and Kitchenham, B., 'Effort estimation using analogy', Proceedings of the 18th International conference on software engineering, pp. 170-178, Berlin, 25-30 of March, 1996

159. Slomka, F., Dorfel, M., Munzenberger, R., Hofmann, R., ''Hardware/Software Codesign and Rapid Prototyping of Embedded systems'', IEEE Design and Test of Computers, vol. 17, no. 3, pp. 28-38, April – June 2000

160. Soklic, M. E., ‘Laboratory for Real-time and Embedded Systems’, Computers in Education Journal, pp. 1–11, vol. 12, no. 4, 2002.

161. Stewart, R.; Wyskidsa, R. & Johannes, J., “Cost Estimator's Reference Manual”, 2nd ed., Wiley Interscience, 1995.

162. Stutzke, R. D., ‘Software estimating technology: A Survey’, Crosstalk: The journal of Defense Software engineering, May 1996 issue.

163. Sztipanovits, J., and Karsai, G., ‘Embedded Software: Challenges and opportunities’, Embedded Software Conference, 2001

164. Thiry, O., and Claesen, L., ‘’A formal verification technique for embedded software’’, IEEE International Conference on Computer Design, pp. 352-357, 7-9 of October, 1996

165. Tran-Cao, D., Abran, A., and Levesque, G., ‘’Functional Complexity Measurement’’, International Workshop on Software Measurement (IWSM 01), Montreal, Quebec, Canada, pp. 232-239, 28-29 of August 2001

166. Vahid, F., and Givargis, T., ‘’Embedded System Design: A Unified Hardware/Software Introduction’’, John Willey and Sons Inc, 2002

167. Vidger, M. R. and Kark, A. W., 1994. ‘Software cost estimation and control’, National Research Council of Canada Software Engineering Report, NRC No. 37116, February 1994

168. Villar, E., and Veiga, M., ‘Embedded System Specification’. In Advanced Techniques for embedded systems design and test, J.C. Lopez et al (eds.), Kluwer Academic Publishers, 1998

169. Vitaliano, W. J., ‘Three design to cost myths’, In: 1994 International conference of the society of American value engineers (SAVE), New Orleans, USA. SAVE annual proceedings, 1996.

170. Voget, S., ‘Development of Product Line Architectures for the Automotive Domain’, International Colloquium at the University of Kaiserslautern - Software Reuse - Requirements, Technologies and Application, Kaiserslautern, Germany, March 2003a.

171. Voget, S., ‘Future Trends in Software Architectures for Automotive Systems’, Advanced Microsystems for Automotive Applications, 7th International Conference on Advanced Microsystems for Automotive Applications, 22-23 of May, Berlin, Germany, 2003b.

172. Voros, N.S., Sanchez, L., Alonso, A., Birbas, A., Birbas, M., ans Jerraya, A., ‘Hardware-Software codesign of complex embedded systems: an approach using efficient process models, multiple formalism specification and validation via co-simulation’. Design Automation for embedded systems, Kluwer Academic Publishers, vol. 8, no. 1, pp. 5-49, 2003

173. Wagner, F. R., Cesario, W. O., Caro, L., and Jerraya, A. A., ‘’Strategies for the integration of Hardware and Software IP components in embedded systems-on-chip’’, Integration, the VLSI Journal, vol37, no. 4, pp. 223 – 252, 2004

174. Williamson, D., ‘Cost and management accounting’, Prentice Hall Europe, London, 1996.

175. Walkerden, F., and Jeffery, R., 1996. ‘Software cost estimation: a review of models, processes and practice’, Center for advanced empirical software research (CAESAR) technical report, #96/01

176. Wilmshurst, T., ‘An introduction to the design of small scale embedded systems’. Palgrave publishing, 2001

177. Wirth, N., ‘Embedded systems and real time programming’. Embedded Software Conference (EMSOFT) Proceedings, pp.486-492, 2001

178. Wolf, W. H., ‘’Hardware-Software Co-Design of Embedded Systems’’, Proceedings of the IEEE, vol. 82, no. 7, pp. 967-989, July 1994

179. www.ilogix.com, ‘State Modelling using Statecharts and timing diagrams’. Online white paper, by B. P. Douglass, accessed 18th of August 2003

180. www.dsysd.com, ‘An Automotive Development Process for Distributed Systems’. Online presentation, accessed 18th of August 2003

181. Yen Yen, T., and Wolf, W., 'Hardware – Software Co-Synthesis of distributed embedded systems'. Kluwer Academic Publishers, April 1996

182. Zhang, Y.F., 'Feature-based cost estimation for packaging products using neural networks', In: Computers in industry, v.32, pp. 95-113.

183. Ziebart, W., 'Car Electronics: Key factors for success for the 90's', 8th International conference on automotive electronics,pp. 1-6, October 28-31, 1991

Appendices

Appendix A : AS - IS Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Mr. Aitor Sarasua Echeverria - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor
Dr. Victor Taratoukhine - Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
a.sarasua@cranfield.ac.uk
r.roy@cranfield.ac.uk
v.v.taratoukhine@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the Cranfield University's Research team as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researchers elicit the experts' knowledge regarding electronic items cost estimating process. Doing this, the researchers will be able to develop a broad view regarding the research domain, identify the problems, threads and opportunities associated with the electronic items cost estimating process and also, identify best practices followed across companies and industries.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position
A1:

Q2: How long have you been holding this position?
A2:

Q3: How many years of experience do you have in this position?
A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?
A4:

Section 2: Questionnaire

Q1: What are the cases which an electronic items cost estimation is produced for?
A1:

Q2: Could you please explain the fundamental stages you go through when you develop an electronic items cost estimation?
A2:

Q3: During the electronic items cost estimation process, is there any input received by any other company department? If yes, what other departments are involved?
A3:

Q4: How is all this information linked?
A4:

Q5: Could you please outline the problems you are currently facing during the electronic items cost estimation process?
A5:

Q6: How do you assess D&D cost?
A6:

Q7: How do you assess the HW D&D cost?

A7:

Q8: How do you assess the SW D&D cost?

A8:

Q9: What additional data/information/method could improve the quality of the electronic items cost estimation process?

A9:

Please use the following space to add any additional information that you feel may be relevant:

Appendix B : Head of Electronics Design Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researchers to elicit the expert's knowledge about D&D an electronic item. Doing this, the researchers will be able to develop a broad view regarding the research domain and identify the problems, threads and opportunities associated with the electronic items D&D.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

<u>Section 1: General Issues</u>

Q1: Please state your name and your job position
A1:

Q2: How long have you been holding this position?
A2:

Q3: How many years of experience do you have in this position?
A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?
A4:

<u>Section 2: Questionnaire</u>

Q1: Please describe the electronics D&D process within your organisation.
A1:

Q2: What are the factors that affect this process?
A2:

Q3: What are the problems this process presents?
A3:

Q4: What are the factors that cause these problems?
A4:

Q5: Literature suggests that in order to D&D an electronic item from scratch, 15% of the D&D effort is devoted to HW D&D, 35% to SW D&D and the rest 50% is consumed in Integration (SW with SW and SW with HW) and Testing (of the complete electronic item). Do you agree with the above opinion?
A5:

Please use the following space to add any additional information that you feel may be relevant:

Appendix C : D&D Elicitation Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the Cranfield University's Research team as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researchers elicit the experts' knowledge regarding electronic items D&D cost estimating process. Doing this, the researchers will be able to develop a broad view regarding the research domain, identify the problems, threads and opportunities associated with the electronic items D&D cost estimating process and also, identify best practices followed across companies and industries.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

Q1: Please distribute 100% of electronics design and development effort between SW D&D, HW D&D and Integration and Testing (Intres), for an item D&D from scratch.

A1:

Q2: How do you currently assess HW D&D effort?

A2:

Q3: How do you assess HW Complexity?

A3:

Q4: Can HW D&D be judged based on the item’s specifications?

A4:

Q5: What are the problems with HW D&D effort estimation?

A5:

Q6: How do you currently assess SW D&D effort?

A6:

Q7: Can SW D&D be judged based on SW's specifications?

A7:

Q8: What are the problems with SW D&D effort estimation?
A8:

Q9: Which are the issues that affect reuse?

A9:

Q10: Can reuse be predicted?

A10:

Q11: Which are the issues that affect integration and testing?
A11:

Q12: Can integration and testing be predicted?

A12:

Q13: When Integration and Testing results are not satisfactory and the item has to be re-examined, what is the system's area that is affected the most?

A13:

Please use the following space to add any additional information that you feel may be relevant:

Appendix D : Statecharts Complexity Metric Development Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researchers validate a proposed list of factors as factors which express Statecharts specification complexity and assign weights to them. Doing this, the researchers will be able to develop a metric for estimating D&D effort for embedded SW, based on Statecharts specifications.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position
A1:

Q2: How long have you been holding this position?
A2:

Q3: How many years of experience do you have in this position?
A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?
A4:

Section 2: Questionnaire

The following list of factors is suggested as a list of complexity factors for Statecharts specifications.

1. Number of states
2. Number of events, variables and activities
3. Number of actions
4. Number of transitions
5. Number of levels in hierarchy
6. Number of parallel machines
7. Number of data types (integer, boolean, etc)
8. Number of mini-specs
9. Number of truth tables

Q1: According to your opinion, is the above presented list complete?
A1:

Q2: Could you please distribute 100% weight on factors 5-9 according to their importance on the complexity of SW development?
A2:

Please use the following space to add any additional information that you feel may be relevant:

Appendix E : SW Metrics and Results Validation Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....
Title: ....
Organisation: ....
Department: ....
Address: ....
....
....
Telephone Number: ....
Fax Number: ....
Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher validate the presented to experts results and metrics. This will ensure that the approach taken is valid and the proposed metrics could be used for embedded SW cost estimation.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

Q1: Is the assumption that the functionality captured in a UML Use Case diagram is attributable to SW correct?

A1:

Q2: Does Productivity change across different application areas and/or domains?

A2:

Q3: Are the results realistic in terms of effort?

A3:

Q4: Is there anything not covered from the model?

A4:

Q5: Are the values assigned to factors realistic?

A5:

Q6: How is this model different than the one you currently use?

A6:

Q7: What are the potential issues in implementing such a methodology?
A7:

Please use the following space to add any additional information that you feel may be relevant:

Appendix F : Complexity Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the Cranfield University's Research team as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher elicit the experts' knowledge on deriving HW D&D complexity values. Doing this, the researcher will be able to use this complexity value on developing a framework for HW D&D cost estimation.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

The design and development effort for the HW part of the embedded systems depends on how complex the system to be deigned is. Therefore, if we want to create a model to estimate the HW D&D effort we have first of all to find a way to assess the system's complexity.

Through our research and expert opinion, the following table has been created:

Driver	Name	Weight	Value	Ftr.
d_1	*Type of Components*		1-10 ICs No micros	0,4
			1-20 ICs No micros	0,6
			1-20 ICs 1 micro	0,8
			More than 20 ICs 2 micros	1
d_2	*No. Components*		0 - 20	0,2
			21 - 50	0,4
			51 - 100	0,6
			101 - 300	0,8
			More than 300	1
d_3	*Memory type*		ROM	0,2
			PROM	0,3
			EPROM	0,4
			RAM	0,5
			OTP	0,8
			FLASH	1
d_4	*Memory Size*		8kb	0,3
			16kb	0,4
			32kb	0,5
			64kb	0,6
			128kb	0,7
			256kb	0,8
			512kb	1
d_5	*No. of Interfaces*		0 to 10	0,2
			10 to 20	0,5
			20 to 30	0,8
			More than 30	1
d_6	*Type of Interfaces*		Digital Slow	0,2
			Digital Quick	0,3
			Communication Bus (LAN, CAN, etc)	0,4
			Differential Inputs	0,6
			Ground Connections	0.2
			Combination of the above	1
d_7	*Functionality Class*		A (Radio)	0,3
			B (Immobiliser)	0,7
			C (ABS)	1
d_8	*Distributed Functionality*		70% distributed	0,3
			40% to 70% distributed	0,6
			Less than 40% distributed	1
d_9	*Test/Acceptance Crit.*		Normal requirements	0,1
			Special Mech. Requirements	0,2
			Special Temp. Requirements	0,4
			Special EMC Requirements	0,4
			Both Temp. and Mech. Requirements	0,6
			Both EMC and Mech. Requirements	0,6
			Both Temp. and EMC Requirements	0,8
			All of them	1

The first two columns, ''Driver'' and ''Name'', describe the driver's number and name. In the third column, ''Weight'', the corresponding weight for that factor needs to be entered. In the fourth column, ''Value'', each of the complexity drivers has been further broken down in sub-categories and each of these sub-categories has been allocated a corresponding ''adjustment factor'' (found in the fifth column, ''Ftr.''). These subcategories and their corresponding adjustment factors have been introduced by the researcher - based on his understanding - to account for the fluctuating level of complexity within the drivers themselves.

For each HW D&D case to be examined, a complexity value will be obtained using the following formula:

$$C = d_1 \cdot r_1 + d_2 \cdot r_2 + d_3 \cdot r_3 + \ldots d_n \cdot r_n = \sum_1^n d_n \cdot r_n \quad \text{(eq. 1)}$$

where C is the complexity value for that specific HW case study, d_n is the weighting of each driver, r_n is the driver's adjustment factor (to account for the driver's internal complexity level) and n is the number of drivers.

<u>Section 3:</u>

Q1: Do you agree with the rationale of the suggested model?
A1:

Q2: Do you consider the list of factors and the sub-categories complete?
A2:

Q3: Do you agree with the adjustment factors allocated to the sub-categories?

A3:

Q4: Would it be possible to distribute 100% weight in the factors d_1 to d_9 of the previous table?

A4:

Please use the following space to add any additional information that you feel may be relevant:

Appendix G : HW D&D Cost Estimation Framework Validation Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher validate the presented to experts HW D&D cost estimation framework. This will ensure that the approach taken is valid and the proposed metrics could be used for embedded HW cost estimation.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

Q1: Do you agree with the rational of the suggested framework?

A1:

Q2: Are the results logical?

A2:

Q3: Could the suggested framework be used to predict embedded HW D&D effort based on the HW D&D complexity?

A3:

Q4: Is there anything not covered from the model?

A4:

Q5: How is this model different than the one you currently use?

A5:

Q6: What are the potential issues in implementing such a methodology?

A6:

Please use the following space to add any additional information that you feel may be relevant:

Appendix H : Reuse Estimation Framework Validation Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher validate the presented to experts Reuse estimation framework. This will ensure that the approach taken is valid and the proposed metrics could be used for embedded HW and SW reuse estimation.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

Q1: Do you agree with the rational of the suggested framework?

A1:

Q2: Could the suggested framework be used to estimate reuse?

A2:

Q3: Could you please give your percentages for each of the 3 cases?

A3:

Case study 1	Question 1: New/existing supplier?		Question 2: New/existing platform?		Question 3: Designed for someone else?	
	HW	SW	HW	SW	HW	SW
Low						
Medium						
High						

Case study 2	Question 1: New/existing supplier?		Question 2: New/existing platform?		Question 3: Designed for someone else?	
	HW	SW	HW	SW	HW	SW
Low						
Medium						
High						

Case study 3	Question 1: New/existing supplier?		Question 2: New/existing platform?		Question 3: Designed for someone else?	
	HW	SW	HW	SW	HW	SW
Low						
Medium						
High						

Q4: Is there anything not covered from the model?

A4:

Q5: How is this model different than the one you currently use?
A5:

Q6: What are the potential issues in implementing such a methodology?

A6:

Please use the following space to add any additional information that you feel may be relevant:

Appendix I : Integration Brake – Down Approach Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher validate the presented to experts Integration Assessment approach. This will ensure that the approach taken is valid and it can improve embedded systems' integration stage visibility and make it more structured and standardised.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

The matrix displayed below presents a breakdown structure on Embedded Systems Integration.

Q1: Do you agree with the Integration categories and sub-categories as presented in the Integration breakdown matrix?
A1:

Q2: Could you please give your percentages for each of the Integration 3 cases and for their sub-categories?
A2:

<table>
<tr><td></td><td colspan="3">Integration Brake – Down Table</td></tr>
<tr><td>Category</td><td>SW to SW</td><td>SW to HW</td><td>In the car</td></tr>
<tr><td>Category's percentage</td><td></td><td></td><td></td></tr>
<tr><td rowspan="5">Subcategories and their percentage</td><td>application SW with:

- RTOS</td><td rowspan="5">''complete'' SW to underlying HW:

- Interface Development:</td><td rowspan="5">complete embedded system into the car's electronic architecture:

- Interconnection of the embedded system with other systems developed by different suppliers:</td></tr>
<tr><td>- Drivers</td></tr>
<tr><td>- FNOS</td></tr>
<tr><td>- Libraries</td></tr>
<tr><td>- Other</td></tr>
</table>

Q3: Any additional comment?
A5:

Please use the following space to add any additional information that you feel may be relevant:

Appendix J : Reuse Estimation Validation Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher validate the presented to experts Reuse estimation framework. This will ensure that the approach taken is valid and the proposed approach could be used for embedded HW and SW Reuse estimation.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

Q1: Do you agree with the rational of the suggested framework?

A1:

Q2: Are the results logical?

A2:

Q3: Could the suggested framework be used to predict Reuse (for both SW/HW)?

A3:

Q4: Is there anything not covered from the model?

A4:

Q5: How is this model different than the one you currently use?

A5:

Q6: What are the potential issues in implementing such a methodology?

A6:

Please use the following space to add any additional information that you feel may be relevant:

Appendix K : Integration Validation Questionnaire

Electronic Parts Design and Manufacturing Cost Modelling

E-Mode Project

Project Duration: November 2002 - October 2005

Electronic Parts Cost Estimating

Mr. Nikos Giannopoulos - Investigator
Dr. Rajkumar Roy - Principal Investigator and Supervisor

Enterprise Integration,
Cranfield University, Cranfield,
BedCompany 1 shire, MK43 OAL, UK.
Tel: +44 (0) 1234 754194,
Fax: +44 (0) 1234 750852
n.giannopoulos@cranfield.ac.uk
r.roy@cranfield.ac.uk

Name: ....

Title: ....

Organisation: ....

Department: ....

Address: ....

....

....

Telephone Number: ....

Fax Number: ....

Email: ....

The information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

INTRODUCTION

This Questionnaire has been developed by the researcher as a tool for supporting the research in the framework of the E-Mode project. E-Mode project aims to produce a better understanding for estimating the cost for the Design and Development (D&D) of automotive electronics.

The aim of this questionnaire is to help the researcher validate the presented to experts Integration Assessment approach. This will ensure that the approach taken is valid and it can improve embedded systems' integration stage visibility and make it more structured and standardised.

Thank you for answering this questionnaire. No individual results will be disclosed to third parties; the information provided will be held in the strictest of confidence, desensitised, and used for academic and research purposes ONLY.

Section 1: General Issues

Q1: Please state your name and your job position

A1:

Q2: How long have you been holding this position?

A2:

Q3: How many years of experience do you have in this position?

A3:

Q4: What is your experience (Academic & Industrial) prior to your current position?

A4:

Section 2: Questionnaire

Q1: Do you agree with the rational of the suggested approach?

A1:

Q2: Are the results logical?

A2:

	Integration Brake – Down Table		
Category	**SW to SW**	**SW to HW**	**In the car**
Category's percentage	30%	20%	50%
Subcategories and their percentage	application SW with: - RTOS	''complete'' SW to underlying HW:	complete embedded system into the car's electronic architecture:
	- Drivers		
	- FNOS	- Interface Development:	- Interconnection of the embedded system with other systems developed by different suppliers:
	- Libraries		
	- Other		

Q3: How is this model different than the one you currently use?

A3:

Q4: Is there anything not covered from the suggested approach?

A4:

Q5: Any additional comment?

A5:

Please use the following space to add any additional information that you feel may be relevant:

Appendix L : Transcript

Head of Electronics Design Interview

Q1: Please describe the electronics D&D process within your organisation.
A1: The ES D&D process in our company follows the same steps with automotive, namely "the V – cycle".

Q2: What are the factors that affect this process?
A2: The Engineers' and experts' knowledge, the policy of our company, the selection of specific items for specific purposes, the tighter constraints than in automotive.

Q3: What are the problems this process presents?
A3: There are 2 main sources of problems.

1. Integration of HW with SW and of the ES within the system's greater architecture
2. Partitioning between HW and SW

Q4: What are the factors that cause these problems?
A4:

1. Integration of HW with SW and of the ES within the system's greater architecture:
    1.1 For SW with HW: This happens because the developers develop the SW code so as to create the desired functionality. However, little or no consideration is given on how this SW will 'seat' on HW, or even if HW is capable of "hosting" the SW. In addition, HW and SW are developed separately, by people who either understand HW or understand SW, but not both.
    1.2 For Integrating the ES within the system's greater architecture: Even if an ES is tested and found to be performing as it was indented to, this does not guarantee that it will do the same when it is integrated in the system's greater architecture. This happens because by developing an ES as a stand-alone system, we loose the interactions between this ES and the overall system. We do test ES using their mathematical modeling but we use system data from past projects. This means that although we discover a lot of mistakes before building the first prototype, we will

not have the real picture until building and testing the first prototype.

2. Partitioning between HW and SW: by the independent development of SW and HW, the "design space" (the overall set of possible acceptable combinations of HW-SW implementations) is seriously limited; we are therefore not sure if we have chosen the optimum solution. In addition, by this separate development there is always the danger of developing SW that will not fit in HW and not discover this until a late stage.

Q5: Literature suggests that in order to D&D an electronic item from scratch, 15% of the D&D effort is devoted to HW D&D, 35% to SW D&D and the rest 50% is consumed in Integration (SW with SW and SW with HW) and Testing (of the complete electronic item). Do you agree with the above opinion?
A5:
Yes, I do agree. The only difference is that in our company, Integration is call "proving" since the aim of this phase is to prove that the ES is working OK.

Please use the following space to add any additional information that you feel may be relevant:

- The effort to D&D an electronic item in the aerospace/defence industry is bigger than in automotive due to (a) stricter and more extensive testing (if an ABS system fails, the driver can still brake, if however the Flight Control fails the consequences could be fatal), and (b) to the selection of components. For example, a fight airplane can experience acceleration up to 10g (10 times the earth's gravitational force). Selected components should be able to undertake this acceleration and accompanied vibration.
- Because of the extensive cost for building and testing a big number of prototypes in order to test an electronic item, testing and validation is being performed using the system's functional mathematical model supported by system test data coming from previous system(s) tests.

D&D Elicitation Questionnaire
(Effort Capture Workshop)

Q1: Please distribute 100% of electronics design and development effort between SW D&D, HW D&D and Integration and Testing (Intres), for an item D&D from scratch.

A1:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
HW:15% SW: 35% (Integration of application SW with other SW) Intres: 50% (integrating SW with HW, integration to car's electronic architecture and system testing)	HW: 15% SW: 30% (integration of application SW with EFNOS, LANs, etc) Intres: 55% (integrating SW with HW, integration to car's electronic architecture and system testing)	HW: 15% SW: 35% (application D&D) Intres: 50% (integrating application SW with other SW (EFNOS, LANs, etc), integration with HW, integration to car's electronic architecture and system testing)	HW: 10% (a lot of copy-paste from libraries) SW: 25% (application D&D and application's calibration) Intres: 65% (integrating application SW with other SW (EFNOS, LANs, etc), integration with HW, integration to car's electronic architecture and system testing)

Q2: How do you currently assess HW D&D effort?

A2:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
By looking at the ES implementation details and using our experience from other similar items in order to understand how different (more or less complex) this is.	Using a combination of the system's implementation details and our experience	We use our experience to understand its complexity in comparison with past projects	Based on our experience to understand how it works, plus assessing its complexity

Q3: How do you assess HW Complexity?

A3:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
Based on the number of Memory type and size, number of components, number and type of interfaces and by comparing with other past projects.	Assessing its Functionality class, distributed functionality and number/type of components in comparison with other, similar projects.	Based on the number and type of components, how distributed its functionality is and on what tests have to be taken.	Using a number of factors (type and size of memory, total number of components and how distributed its functionality is.

Q4: Can HW D&D be judged based on the item's specifications?

A4:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
No, specs say 'what' the item should do, not 'how' it should do it. And this 'what' can be realised through a big number of alternative implementations.	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1

Q5: What are the problems with HW D&D effort estimation?
A5:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
It always depends on the case. As said earlier, you can not judge based on the specifications. On the other hand, neither BOM is convenient, because there are parts that are hidden inside the item or items we have no knowledge of (like ASICs).	Specifications (as explained earlier) and BOM. The problem with BOM is that by comparing two items, you are deriving historical and not D&D cost.	Apart from the specifications problem, the other problem is BOM, because of the hidden items that an ES holds, or because of items with protected by the supplier with IP rights.	Specifications and BOM for all the reasons that were raised by the others.

Q6: How do you currently assess SW D&D effort?

A6:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
As an overhead (fixed percentage) on top of the manufacturing cost	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1

Q7: Can SW D&D be judged based on SW's specifications?

A7:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
No, specs say 'what' the item should do, not 'how' it should do it. And this 'what' can be realised through a big number of alternative ways.	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1

Q8: What are the problems with SW D&D effort estimation?
A8:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
That the necessary information (ie code, flow diagrams, etc) is protected by IP rights.	The same as Engineer 1	The same as Engineer 1	The same as Engineer 1

Q9: Which are the issues that affect reuse?

A9:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
Even in situations we know there is reused involved, this is very difficult to be realised, since the item's details are protected by supplier's IP	The fact we do not have access to the SW or sometimes even in HW	The lack of information due to supplier's IP	The fact that most information on the item's implementation are hidden and protected by supplier's IP

Q10: Can reuse be predicted?

A10:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
No, because there is no access to information. Even if we are sure there is any amount of reuse, we can not prove it.	Even if we are 100% sure that reuse is involved, there is no way of accessing how much reuse has been applied. This is because the necessary details are protected.	No, it can not predicted because we do not have access to the implementation details and there is no way of finding out how much of this has been reused and where.	Reuse is hidden by the supplier. Even if we know it has been developed for someone else, the supplier says: 'I wasn't paid the D&D cost by the other company, so I now need my money back'. We do use expert judgement to derive an indication of reuse, but again, we can not prove it.

Q11: Which are the issues that affect integration and testing?
A11:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
It depends on the SW and HW to be integrated and what will happen when the item is tested on real situations.	Depends on each case from the SW and HW to be integrated	On the SW and HW to be integrated plus the calibration that maybe needed after testing	The item's SW and HW, the side effects on the car and calibration.

Q12: Can integration and testing be predicted?

A12:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
It's a case by case situation	No, because you never know what will happen when the item will be integrated in the car	It's case by case, because you can never predict the side effects of the SW-SW, SW-HW and into the car integration	It's case by case

Q13: When Integration and Testing results are not satisfactory and the item has to be re-examined, what is the system's area that is affected the most?

A13:

Engineer 1	Engineer 2	Engineer 3	Engineer 4
SW, because it is much easier to be modified	SW, because in this late stage it is not easy to redesign the HW	SW, because code can be modified but it is not easy to redevelop HW	SW because it can change easier than HW

HW Complexity Questionnaire

The design and development effort for the HW part of the embedded systems depends on how complex the system to be deigned is. Therefore, if we want to create a model to estimate the HW D&D effort we have first of all to find a way to assess the system's complexity. Through our research and expert opinion, the following table has been created:

Driver	Name	Weight	Value	Ftr.
d_1	*Type of Components*		1-10 ICs No micros	0,4
			1-20 ICs No micros	0,6
			1-20 ICs 1 micro	0,8
			More than 20 ICs 2 micros	1
d_2	*No. Components*		0 - 20	0,2
			21 - 50	0,4
			51 - 100	0,6
			101 - 300	0,8
			More than 300	1
d_3	*Memory type*		ROM	0,2
			PROM	0,3
			EPROM	0,4
			RAM	0,5
			OTP	0,8
			FLASH	1
d_4	*Memory Size*		8kb	0,3
			16kb	0,4
			32kb	0,5
			64kb	0,6
			128kb	0,7
			256kb	0,8
			512kb	1
d_5	*No. of Interfaces*		0 to 10	0,2
			10 to 20	0,5
			20 to 30	0,8
			More than 30	1
d_6	*Type of Interfaces*		Digital Slow	0,2
			Digital Quick	0,3
			Communication Bus (LAN, CAN, etc)	0,4
			Differential Inputs	0,6
			Ground Connections	0.2
			Combination of the above	1
d_7	*Functionality Class*		A (Radio)	0,3
			B (Immobiliser)	0,7
			C (ABS)	1
d_8	*Distributed Functionality*		70% distributed	0,3
			40% to 70% distributed	0,6
			Less than 40% distributed	1
d_9	*Test/Acceptance Crit.*		Normal requirements	0,1
			Special Mech. Requirements	0,2
			Special Temp. Requirements	0,4
			Special EMC Requirements	0,4
			Both Temp. and Mech. Requirements	0,6
			Both EMC and Mech. Requirements	0,6
			Both Temp. and EMC Requirements	0,8
			All of them	1

The first two columns, ''Driver'' and ''Name'', describe the driver's number and name. In the third column, ''Weight'', the corresponding weight for that factor needs to be entered. In the fourth column, ''Value'', each of the complexity drivers has been further broken down in sub-categories and each of these sub-categories has been allocated a corresponding ''adjustment factor'' (found in the fifth column, ''Ftr.''). These subcategories and their corresponding adjustment factors have been introduced by the researcher - based on his understanding - to account for the fluctuating level of complexity within the drivers themselves.

For each HW D&D case to be examined, a complexity value will be obtained using the following formula:

$$C = d_1 \cdot r_1 + d_2 \cdot r_2 + d_3 \cdot r_3 + ... d_n \cdot r_n = \sum_{1}^{n} d_n \cdot r_n \quad \text{(eq. 1)}$$

where C is the complexity value for that specific HW case study, d_n is the weighting of each driver, r_n is the driver's adjustment factor (to account for the driver's internal complexity level) and n is the number of drivers.

Section 3:

Q1: Do you agree with the rationale of the suggested model?
A1:

Expert 1: Yes, I do. This is a good idea for describing an ES complexity
Expert 2: Yes, it is a sound approach for estimating HW complexity
Expert 3: Yes, I do. It is a structured approach towards assessing complexity

Q2: Do you consider the list of factors and the sub-categories complete?
A2:

Expert 1: The 'Test - Acceptance Criteria' complexity factor should be redesigned to cover protection from contact with any foreign matter and/or water. In the ''Memory Size'' factor, there should be also the subcategory of 1MB memory size included.
Expert 2: In addition to Expert's 1 comments, the 'Test - Acceptance Criteria' complexity factor should also cover extreme temperature requirements (>85^{0}C).
Expert 3: In the ''Memory Size'' factor, there should be also the subcategory of 1MB memory size included.

Q3: Do you agree with the adjustment factors allocated to the sub-categories?

A3:

Expert 2 and Expert 3: In the ''Distributed Functionality'' factor, the percentages assigned to the subcategories should be the other way around: 70% and more distributed functionality should have an adjustment factor of 1, 40% to 70% an adjustment factor of 0,6 and for distributed functionality less than 40%, an adjustment factor of 0,3.

Q4: Would it be possible to distribute 100% weight in the factors d_1 to d_9 of the previous table?

A4:

Weights				
Driver	**Name**	**Expert 1**	**Expert 2**	**Expert 3**
d_1	Type of Components	20	20	20
d_2	Number of Components	10	10	10
d_3	Memory type	5	5	5
d_4	Memory Size	5	10	10
d_5	Number of Interfaces	10	10	10
d_6	Type of Interfaces	20	15	20
d_7	Functionality Class	10	10	10
d_8	Distributed Functionality	15	10	10
d_9	Test/Acceptance Criteria	5	10	5

HW D&D Cost Estimation Framework Validation Questionnaire

Q1: Do you agree with the rational of the suggested framework?

A1:

Expert 1: Yes, I do. It is a logical sequence of steps for deriving an estimation for HW D&D cost.
Expert 2: Yes, this approach creates a sound structure for assessing HW D&D cost

Q2: Are the results logical?

A2:

Expert 1: As shown by the correlation coefficient, yes, they are.
Expert 2: There seems to be a good amount of correlation

Q3: Could the suggested framework be used to predict embedded HW D&D effort based on the HW D&D complexity?
A3:

Expert 1: Yes, it could, but for the body domain only
Expert 2: Yes, it could

Q4: Is there anything not covered from the model?

A4:

Expert 1 and Expert 2: Infotainment and Chassis

Q5: How is this model different than the one you currently use?
A5:

Expert 1 and Expert 2: There is currently no model used for HW D&D effort estimation within the organisation.

Q6: What are the potential issues in implementing such a methodology?

A6:

Expert 1: Data (cost, complexity factors, etc) have to be collected and used to further calibrate the model. In addition, the model has to be expanded to the Infotainment and Chassis domains.
Expert 2: More than 6 case studies have to be obtained and used in the model development for better adjustment of the correlation coefficient.

Statecharts Complexity Metric Development

The following list of factors is suggested as a list of complexity factors for Statecharts specifications.

1. Number of states
2. Number of events, variables and activities
3. Number of actions
4. Number of transitions
5. Number of levels in hierarchy
6. Number of parallel machines
7. Number of data types (integer, boolean, etc)
8. Number of mini-specs
9. Number of truth tables

Q1: According to your opinion, is the above presented list complete?

A1:

Experts 1, 2, 3, 4 and 5: The list is complete

Q2: Could you please distribute 100% weight on factors 5-9 according to their importance on the complexity of SW development?

A2:

Expert 1	Number of levels in hierarchy: 35%, Number of parallel machines: 30%, Number of data types, Number of mini-specs and Number of truth tables: 35%
Expert 2	No weighting, it is up to the expert's knowledge to decide
Expert 3	Number of levels in hierarchy: 35%, Number of parallel machines: 30%, Number of data types, Number of mini-specs and Number of truth tables: 35%
Expert 4	Number of levels in hierarchy: 25%, Number of parallel machines: 30%, Number of data types, Number of mini-specs and Number of truth tables: 45%
Expert 5	No weighting, all factors are of equal importance (Therefore, each factor is assigned a weight of 33%)

Reuse Estimation Framework Validation Questionnaire

Q1: Do you agree with the rational of the suggested framework?

A1:

Expert 1: Yes, the framework follows a logical approach.

Expert 2: It is a sound framework and offers a standardised estimation approach.

Q2: Could the suggested framework be used to estimate reuse?

A2:

Expert 1: Yes, it can. It consists a more structured way of estimating reuse.

Expert 2: Yes, since we have no access to any other information.

Q3: Could you please give your percentages for each of the 3 cases?

A3:

	Question 1: Is it going to be designed for a new or an existing platform?				**Question 2: Has it already been designed for someone else?**			
Case Study 1	**Expert 1**		**Expert2**		**Expert 1**		**Expert2**	
	HW	**SW**	**HW**	**SW**	**HW**	**SW**	**HW**	**SW**
Low	10	10	10	10	80	90	80	90
Medium	20	20	20	20	80	90	80	90
High	30	30	80	90	80	90	80	90
Case Study 2	**Expert 1**		**Expert2**		**Expert 1**		**Expert2**	
	HW	**SW**	**HW**	**SW**	**HW**	**SW**	**HW**	**SW**
Low	60	60	60	60	90	75	90	60
Medium	75	75	75	75	90	75	90	60
High	75	75	90	75	90	75	90	75
Case Study 3	**Expert 1**		**Expert2**		**Expert 1**		**Expert2**	
	HW	**SW**	**HW**	**SW**	**HW**	**SW**	**HW**	**SW**
Low	20	20	20	20	95	50	95	50
Medium	30	30	30	30	95	50	95	50
High	50	50	95	50	95	50	95	50

Q4: Is there anything not covered from the model?

A4:

Expert 1: No, I do not think that there is something missing from the model
Expert 2: Considering that necessary information is not available, this offers a good way of estimating percentage of reuse within an ES

Q5: How is this model different than the one you currently use?
A5:

Expert 1: There is currently no model used for reuse estimation within our organisation.
Expert 2: No model is currently used to predict reuse; reuse is estimated based on expert judgment.

Q6: What are the potential issues in implementing such a methodology?

A6:

Expert 1: Data that are now kept in the engineers' draws have to be collected and analysed, so that Reuse percentage matrices to be developed for all the ES
Expert 2: Engineers will have to provide percentages for each entry of the Reuse matrix.

Please use the following space to add any additional information that you feel

may be relevant:

Expert 1: It is not possible to estimate reuse (either for HW or SW) only by the fact that a system would be designed either by a new or by an existing supplier, since the answer to this question only does not supply any important information for the percentages to be derived.
Expert 2: If the item is designed by a new or an existing supplier has a cost impact on the organisation, which is particular to the organisation itself. However, it is not possible for a reuse percentage to be estimated based only on that fact

Appendix M : IDEF

Introduction to $IDEF_0$ Standard**

This standard describes the $IDEF_0$ modelling language (semantics and syntax), and associated rules and techniques, for developing structured graphical representations of a system or enterprise. Use of this standard permits the construction of models comprising system functions (activities, actions, processes, operations), functional relationships, and data (information or objects) that support systems integration. The primary objectives of this standard are:

1. To provide a means for completely and consistently modelling the functions (activities, actions, processes, operations) required by a system or enterprise, and the functional relationships and data (information or objects) that support the integration of those functions.

2. To provide a modelling technique which is independent of Computer-Aided Software Engineering (CASE) methods or tools, but which can be used in conjunction with those methods or tools.

3. To provide a modelling technique that has the following characteristics:

    1. Generic (for analysis of systems of varying purpose, scope and complexity).
    2. Rigorous and precise (for production of correct, usable models).
    3. Concise (to facilitate understanding, communication, consensus and validation).
    4. Conceptual (for representation of functional requirements rather than physical or organizational implementations).
    5. Flexible (to support several phases of the lifecycle of a project).

The use of this standard is strongly recommended for projects that:

** Extracted from: INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0). US Federal Information Processing Standards Publications (FIBS PUBS). Issued by the National Institute of Standards and Technology. 21 December 1993.

1. Require a modelling technique for the analysis, development, re-engineering, integration, or acquisition of information systems.
2. Incorporate a systems or enterprise modelling technique into a business process analysis or software engineering methodology.

The specifications of this standard are applicable when system or enterprise modelling techniques are applied to the following:

1. Projects requiring $IDEF_0$ as the modelling technique.
2. Development of automated software tools implementing the $IDEF_0$ modelling technique.

$IDEF_0$ includes both a definition of a graphical modelling language (syntax and semantics) and a description of a comprehensive methodology for developing models.

$IDEF_0$ may be used to model a wide variety of automated and non-automated systems. For new systems, $IDEF_0$ may be used first to define the requirements and specify the functions, and then to design an implementation that meets the requirements and performs the functions. For existing systems, $IDEF_0$ can be used to analyze the functions the system performs and to record the mechanisms (means) by which these are done.

The result of applying $IDEF_0$ to a system is a model that consists of a hierarchical series of diagrams, text, and glossary cross-referenced to each other. The two primary modelling components are functions (represented on a diagram by boxes) and the data and objects that inter-relate those functions (represented by arrows).

As a function modelling language, $IDEF_0$ has the following characteristics:

1. It is comprehensive and expressive, capable of graphically representing a wide variety of business, manufacturing and other types of enterprise operations to any level of detail.
2. It is a coherent and simple language, providing for rigorous and precise expression, and promoting consistency of usage and interpretation.
3. It enhances communication between systems analysts, developers and users through ease of learning and its emphasis on hierarchical exposition of detail.

4. It is well tested and proven, through many years of use in the US Air Force and other government development projects, and by private industry.

In addition to definition of the $IDEF_0$ language, the $IDEF_0$ methodology also prescribes procedures and techniques for developing and interpreting models, including ones for data gathering, diagram construction, review cycles and documentation.

$IDEF_0$ uses an easy way to represent the different parts involved in a process, which is under study. It is based on linked boxes and arrows going through all the activities and breaking down every task involved in them. The single function represented on the top-level context diagram may be decomposed into its major sub-functions by creating its child diagram. In turn, each of these sub-functions may be decomposed, each creating another, lower-level child diagram.

The syntax of $IDEF_0$ is illustrated in Figure 1, which is composed of these elements:

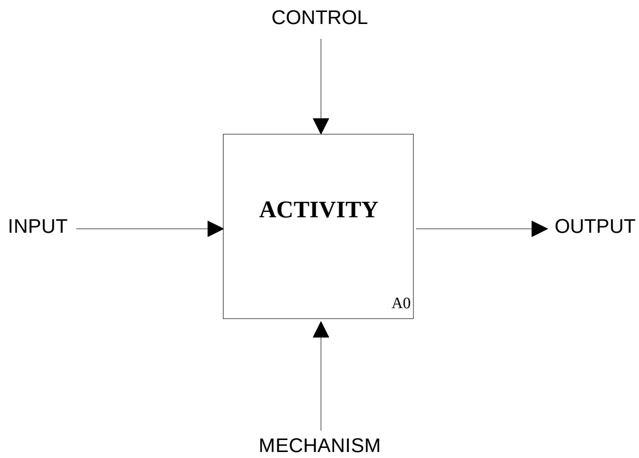


Figure1: $IDEF_0$ Syntax Representation