Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2008
August 3-6, 2008, Brooklyn, New York, USA

DETC2008-49905

A FRAMEWORK FOR THE EVALUATION OF RESOURCE EFFICIENCY IN AUTOMOTIVE EMBEDDED SYSTEMS

Erik Persson
Royal Institute of Technology, KTH
Industrial Engineering & Management
SE-10044 Stockholm, Sweden
erikperaxel@gmail.com

Håkan Gustavsson
Scania CV AB and Mälardalen University
Dept. Computer Science and Electronics
SE-72123 Västerås, Sweden
Hakan.Gustavsson@mdh.se

ABSTRACT

This article discusses the resource utilization of embedded systems in the automotive industry. Traditionally, the major cost driver – or resource input – has been regarded as the hardware cost. Issues such as software development costs and maintenance costs have historically been neglected. In order to address this, the article embraces the more comprehensive view on resources that a resource can be regarded as anything which could be thought of as a strength or weakness of a given firm. In this article the major drivers of resource consumption are identified. The work has also included several interviews with employees in order to find empirical data of the embedded systems in vehicles.

This paper proposes a method to evaluate the resource efficiency of user functions implemented through the embedded system. By the use of Data Envelopment Analysis – which has proven to be a useful method – the resource utilization of six user functions is evaluated. Future work of particular interest would be to perform a more extensive case study.

1 INTRODUCTION

Historically, there has been a focus on hardware costs in the automotive industry whereas the cost of software has been neglected, or at least considered hard to estimate and thus often overlooked. This is something pointed out in a report by McKinsey & Company, where it is stated that automotive players still view hardware as their main differentiating factor, and that software on the other hand is viewed as necessary but easy to change and free of cost [1].

Today most innovations made within the automotive domain are driven by electronics. A study made by Mercer Management Consulting and Hypovereinsbank in 2001 [2] claims that the total value of software in cars will rise from 4% to 13% by 2010.

Looking at resources from only a traditional hardware perspective is limiting. The next step, so to speak, would be a more integrated view on the ECU and the embedded system: to consider the spending on not only hardware but also on software. However, this view is also limited in the sense that it looks at the embedded system as an isolated entity, and does not take into account the implications for the resources of the company.

The idea of looking at firms as a broader set of resources goes back to the work of E.T. Penrose and her book “The theory of the growth of the firm” from 1957 [13], a book that has laid the foundation for the more recent “resource-based view” of firms.

It is of importance to be able to quantify the degree of cost-efficiency of a solution and its resource utilization. One reason to this is that it facilitates the evaluation and comparison of different design solutions and makes it possible to better value the resources that are consumed by the system. In order to make an adequate design decision, one must consider numerous factors. There are obvious aspects such as size, cost and capacity of a component, yet other less tangible factors are very important, factors such as customer preferences, development cost, production volume and time to market. All these factors – and many more – influence the necessary input of resources as well as the magnitude of the output, in other words, these factors affect how well the system is being utilized.

To address this problem, the following research question was formulated: How can one quantify the resource utilization in automotive embedded systems in the automotive industry?

2 METHOD AND OBJECTIVE

The purpose of this paper is to investigate the economic resource utilization in automotive embedded systems. The initial phase of the work consisted of formulating a problem statement and research questions. The subsequent step was to conduct an initial literature survey to gain further insight in the field, find out the state-of-the-art research and to be able to formulate relevant interview questions. The next step consisted of conducting the interviews. These interviews proved to be not only a way of collecting data, but also they were instrumental in grasping the problem in its context. The extensive literature survey also confirmed that the work on economic resource utilization in embedded systems is scarce in the research community.

Once the theoretical framework was established, a model based on this framework could be created. This model has then been used in the company case study, with empirical data from the company. The studied company is an international well known vehicle manufacturer of commercial vehicles and should be comparable to the rest of the industry. Proceeding from this study, an analysis of the results and the applicability of the model has been formed, providing the conclusions drawn from the work.

3 DATA ENVELOPMENT ANALYSIS

The theoretical framework of Data Envelopment Analysis (DEA) will be used to construct an approach to evaluate resource efficiency in embedded systems. The theory outlined below comes from the books "Data Envelopment Analysis"[4] and "Handbook of Data Envelopment Analysis"[5] both written by Cooper *et al*.

It is common to evaluate the efficiency of for instance a business firm by dividing its output by the corresponding input. The output is the positive outcome, and should generally be as large as possible. The input reflects the effort needed to attain this output, and it should generally be as small as possible. To measure the performance of a company it is very common to use key performance indicators (KPI). Some of these measures follow the definition of efficiency. Examples may be:

- Gross and net margin
- Revenue per employee
- Sales per employee etc.

Efficiency is a measure of performance and it is defined as follows:

$$Efficiency = \frac{Output}{Input}$$

However, the KPIs mentioned above are based on one input and one output (single input and single output). Hence, these measures are often misleading when overall efficiency is to be measured. The improvement of one output may require the increase of an input that is not reflected by the KPI. Consider an increase in the measure "Sales per employee". This measure does not tell us anything about how costly this increase in sales was. Maybe it required extensive investments in the production plant or in terms of marketing?

To try to get around this problem, one uses normally many KPIs to reflect the different aspects of a company. However, a method that can take in several evaluation factors at the same time to measure the efficiency would be desirable. This can be done through DEA.

Data Envelopment Analysis is a relatively new method for measuring and evaluating performance when several inputs and outputs are included in the same measure. It evaluates the performance of a set of peer entities called Decision Making Units (DMUs) which convert multiple inputs into multiple outputs. The definition of a DMU is generic and flexible – it can be a company, a business unit, a hospital or even an ECU. This work uses the original CCR DEA-model which was found to be sufficient.

The efficiency is simply put calculated by dividing output by input. The difference when multiple input and output are used is that they may measure completely different factors in different units – so that the total input and output must be weighted. This is accomplished as the efficiency is calculated as the weighted sum of outputs divided by the weighted sum of inputs.

$$Efficiency = \frac{y_1u_1 + y_2u_2 + ... + y_su_s}{x_1v_1 + x_2v_2 + ... + x_mv_m}$$

Here y_i is output element *i*, x_i is input element *i*, u_i is the weight associated to output *i* and v_i is the weight associated to input *i*. DEA uses linear programming theory to determine the weights associated to each input and output.

A DMU is considered efficient if it exhibits the following properties: the efficiency equals one and all weights are greater than zero. Otherwise the DMU is inefficient.

One of the major benefits of Data Envelopment Analysis is that the operator does not have to determine the weights subjectively and thus the relative importance of different factors. Instead, the model calculates through Linear Programming for each DMU (for instance ECU) the best possible set of weights in order to maximize its efficiency, under the constraint that the efficiency of all the other DMUs does not exceed one. Another benefit is that a mixture of quantitative and qualitative factors can be used and that the units can be different.

4 EVALUATION FRAMEWORK

In the basic version of Data Envelopment Analysis, all criteria are assumed to be of equal importance. However, it is possible to incorporate a priori knowledge such as price information to make sure that the most important criteria are the most influential to the analysis. This is done by adding constraints to the optimization problem. When introducing constraints, it is important that the factors involved by the constraints are measured in the same units, which is the case in this analysis, as all inputs are measured using the same grading scale. The following constraints will be used in the Data Envelopment Analysis:

$$0.25v_{HW} \leq v_{SW} \leq 0.67v_{HW} \quad (1)$$

This constraint implies that the cost of software is estimated to be between 25% and 67% of the cost of hardware. These numbers are based on the study made by Mercer Management and Hypovereinsbank [2]. These numbers apply to passenger cars, however it is well known that that the evolution of commercial vehicles lags the evolution of passenger cars. For instance, Zientz [12] states that truck manufacturers have tended to introduce electronic solutions only when the maturity of the new technology has already been proven by its application on the passenger car market. Hence it is very reasonable to apply this interval in the analysis conducted. A relation between hardware cost and software development cost has already been established. Looking at Figure 1 one can see that independent of time, maintenance and development costs have roughly been equally large. This implies that the maintenance cost should be between 25% and 67% of the cost of hardware.

$$0.25v_{HW} \leq v_{Maint} \leq 0.67v_{HW} \quad (2)$$

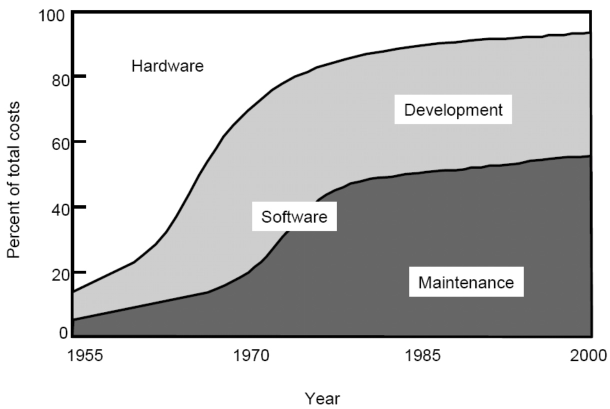

Figure 1 Hardware and software cost trends [7]

4.1 CHOICE OF INPUT

The following evaluation criteria have been identified as the most important:

Hardware cost: This criterion has traditionally been regarded as the far most important cost factor. And it is very important, however it is not the only parameter.

Software development cost: As stated previously there is an increasing importance of software costs.

Maintenance cost: For embedded real-time systems, maintenance costs may be up to four times higher than development costs[9]. According to Fornaciari, development costs and maintenance costs are of approximately the same size [7]. These figures motivate taking this factor into account as well.

Wiring harness cost: During this work a brief survey regarding the cost of wiring harness was conducted. The result was that a very coarse estimate is that the wiring harness cost equals the hardware cost of an ECU.

Time-to-market: In general, a vendor whose product reaches the market quicker than its competitor has a better chance of reaching supremacy in that product group. Debardelaben et.al states the following [6]: “Time-to-market and life cycle costs are key factors in the success of these products in the competitive electronics marketplace. These costs, therefore, should have a dominant influence on the design of embedded microelectronic systems.”

Quality

Quality is one of the most important issues for commercial vehicles. Moreover, quality is the number one feature of the studied company. In this analysis quality costs are seen as opportunity costs, which should be minimized. In DEA, inputs are factors that should be minimized and outputs should be maximized. Hence, quality is regarded as an input.

Hardware resource	Weight
I/O	4
Processor	2
Flash memory	1
EEPROM	1
RAM	1

Figure 2 Assignment of weights and assumptions made

In Figure 2, the assignment of weights is presented. These are estimations based on empirical findings on the prices of these components from the interviews. At the company, products are developed according to the Product Identity. It is divided into two parts, “prestige” and “performance”. Prestige refers to if the product helps to meet the expectations customers have on its products. Performance obviously refers to the performance of the product, and its constituents are listed in the figure below. The measures “Prestige” and “Performance” are assumed equally important. Hence they have a weight relation of 1 to 1.

4.2 CHOICE OF OUTPUT

Revenue: This is the most important output. It captures sales volume and value added.

The approach that will be used in this framework is to use qualitative data. An example of qualitative data is grades. In the true meaning, grades are quantitative data, as the answers have been transformed into numbers. However, in this context they will be referred to as qualitative data. This can be contrasted to for instance sales volume, where absolute data can easily be identified, and need not to be transformed into a relative measure using some grading scale. With such a transformation obviously precision of the data is lost. One of the strengths with Data Envelopment Analysis is that a combination of qualitative and quantitative data can be used.

There are several benefits of using qualitative data. First of all, with this approach it is relatively fast to collect the required and valid data and a questionnaire can be formulated. A proposed questionnaire is presented in Appendix A. Moreover, many factors are intrinsically difficult to estimate, such as the hardware cost of a particular user function. For instance, the processor load due to a particular user function is virtually impossible to measure. In such cases isolating resource consumption drivers and investigating them qualitatively probably gives a more valid result.

Moreover, the same units must be used when assigning weight constraints. Weight constraints are used to weight the relative importance of various factors. It should however be pointed out that it is best to use quantitative data to the furthest extent possible when applying Data Envelopment Analysis. However, in this case, the estimations of quantitative data would not be of a sufficiently high quality, and hence the use of qualitative data would yield more reliable data.

The risk of poor quality estimates is the reason for why the aspects of quality and time to market will not be included in the analysis performed in this paper. In particular, it is difficult to assign these issues to a particular function. However, these are important issues, but as estimations of these costs are expected to be highly unreliable, they will be excluded.

5 THE EVALUATION TOOL

A tool was developed to evaluate resource utilization. The evaluation tool consists of two parts; a basic Microsoft Excel sheet and a Matlab model. All calculations are based on the theoretical framework presented by Cooper[4][5] and the assumptions outlined in this paper.

The first step is to collect the data for the evaluation tool. This is done through the questionnaire (Appendix A), preferably answered by the “function owner”, which is the title of a person responsible for a user function. A user function is at the company referred to as functionality that is unique in the system, is clearly useful to the user as such and is triggered by the user. In general a user function incorporates not just one particular ECU, but an extensive part of the whole electronics system. In other words, in most cases a user function is a distributed function.

The next step is simply to populate the Excel-sheet with the numerical answers to the questions. The model then calculates one single value for each input and output that will be used for the Data Envelopment Analysis. These values are transferred to Matlab where the calculations are made.

5.1 WORK FLOW OF THE ANALYSIS

1. Perform a graphical 3-variable analysis of the data.

Use the output and the two most important inputs as data in order to get an overview. In this case the inputs will be hardware cost and software cost, yielding a total of three variables or evaluation criteria. Software development costs and maintenance costs are assumed equally important, hence their grading is averaged to form a compound software cost. In the Matlab model no weight constraints are introduced at this stage.

2. Calculate the corresponding efficiency scores analytically and compare with the graphical result.

3. Extend the analysis to 4-variables or evaluation criteria.

Calculate the efficiency scores analytically and compare to previous results. In this case the software cost will be split into development cost and maintenance cost, in order to provide 4-variables.

4. Introduce the weight constraints.

Calculate the efficiency scores analytically and compare to previous results. In this case there are two weight constraints that will be introduced.

5. Sensitivity analysis

Investigate how much an evaluation criterion must be improved in order for an alternative to become efficient.

5.2 CASE STUDY – USER FUNCTIONS

In this section, six user functions will be evaluated. The questionnaire created has been used to map these fictitious user functions. The user functions where for validated for relevance by industry experts. Only optional user functions have been evaluated. Their sales volume is between 5 000 and 35 000 which are reasonable numbers for optional functions. The basic characteristics of these six functions are as follows:

User function 1: It is not an advanced function and the program code is small, and it requires little ECU hardware resources. The wiring harness cost is on the other hand almost average. Due to the small code size, the software cost is low both regarding maintenance and development. Its valued added is average, however its sales volume is high (25 000).

User function 2: It is an advanced function with a large application code. The wiring harness cost is high. As a consequence, its total hardware cost is high. Due to a large application code and little code reuse, the development cost is high. The maintenance cost is average, as requirements of the function are not very prone to change. Its value added is slightly above average, due to its increase of customer satisfaction, however its sales volume is quite low (8 000).

User function 3: It is not an advanced function and the application code is quite small. On the other hand, it requires much I/O and the wiring harness cost is above average. Hence, the total hardware cost is above average. Due to a small application code the software cost is low both regarding maintenance and development. Its value added is almost average, but its sales volume is very high (30 000).

User function 4: It has an application code that is slightly above average in size. In addition, the required I/O and the wiring harness cost is above average, yielding a total hardware cost slightly above average. As the required reliability is high and little code reuse has been possible, the development cost is above average. However, due to the use of good programming style and documentation, the maintenance cost is average. Its value added is slightly above average, and its sales volume is good (15 000).

User function 5: It is a very advanced function with a large application code. It uses much I/O and a relatively expensive wiring harness. Thus, the total hardware cost is high. Its development cost is high, and the maintenance cost above average. On the other hand its value added is high, as it is a necessary function and it increases the performance. Its sales volume is very high (35 000).

User function 6: It requires advanced calculations, but the I/O and wiring harness cost is below average, yielding an average total hardware cost. The development cost is also average, however the software maintenance cost is quite low. The value added is high, as the function increases the prestige and is good compared to those of competitors. However, its sales volume is low (5 000).

5.3 CASE STUDY

5.3.1 STEP 1 – GRAPHICAL 3-VARIABLE

Software cost is calculated as the average of software development and software maintenance costs in Swedish Krona (SEK). Using the graphical representation of Data Envelopment Analysis yields the following result

	Hardware cost	Software cost	Revenue
Units:	Grade: 1-9	Grade: 1-9	Virtual SEK
UF			
1	3,1	3,4	110 000
2	6,7	5,6	41 600
3	5,3	3,3	126 000
4	5,6	5,7	84 000
5	6,9	6,7	252 000
6	4,6	4,3	34 000

Figure 3 Input and output for step 1

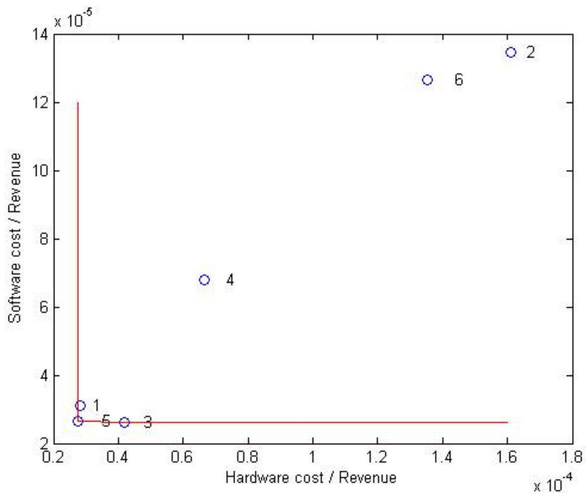

Figure 4 Graphical Data Envelopment Analysis

From the plot it can be seen that the least efficient (basically the further from the efficient frontier the less efficient) user functions are functions 4, 6 and 2, which also corresponds to the user functions with the lowest sales volumes. Functions 1, 5 and 3 are the most efficient. It is clear that user function 1 does not belong to the efficient frontier (even though it is close), and hence cannot be regarded as efficient. A point belonging to the efficient frontier is regarded efficient and the efficient frontier is defined as follows:

There is no point on the frontier line that can improve one of its input values without worsening the other.

At first sight, this leads to the conclusion that function 3 is not efficient. If the line were completely horizontal this would be true. The line connecting user function 5 and 3 is however not horizontal. A more detailed analysis shows that the line has a slightly negative slope. This is also reflected by the weights produced by the analytical analysis. A completely horizontal line is equivalent to a corresponding zero-weight. Looking at the weights, the weight for hardware cost is small, but non-zero.

A problem of this is where to practically draw the line of what is zero. In this case the weight for hardware cost for function 3 is a factor 30 smaller than the weight for function 5. In this case, from a practical perspective, function 3 should not be regarded as efficient. However, in the following analysis this will be disregarded, and the formal definition for efficiency will be employed.

It is straight-forward to explain the relative results for functions 3 and 5. The revenue of user function 5 is roughly twice that of 3. On the other hand, the software cost is just half. However, the hardware cost is more comparable in size of the two. This means that the user functions are comparable with respect to the vertical axis, but function 5 dominates the horizontal axis. If both inputs of function 3 would have been half of those of function 5, then they would have been equally efficient. Please remember that efficiency is a ratio, and that a doubling of the outputs is cancelled by a doubling of the inputs required.

5.3.2 STEP 2 – ANALYTICAL 3-VARIABLE

The analytical Data Envelopment Analysis (without constraints) yields the following efficiency scores:

UF	*Efficiency*	*Efficient*
1	0,97	No
2	0,20	No
3	1,00	Yes
4	0,40	No
5	1,00	Yes
6	0,21	No

Figure 5 Efficiency scores from step 2

As can be seen, the results correspond to those of the graphical analysis. User function 3 and 5 are both efficient; their efficiency score is 100% and all weights are greater than zero. Hence they conform to the definition of being efficient.

This can also be seen by analyzing the graphical version; user function 3 and 5 both lie on the efficient frontier. It should also be noted that user function 1 has a very high efficiency. One may think that the gap of 3% to user function 3 and 5 can be more or less disregarded considering the imprecision of the data employed. However, quite large changes in the data are required in order for user function 1 to become efficient. This is discussed in step 5.

This initial analysis shows that sales volume, which directly affects the revenue, seems to be the major differentiating factor.

5.3.3 STEP 3 – ANALYTICAL 4-VARIABLE

The next step in the analysis is to extend the problem to a 4-variable problem, with three inputs and one output. The additional input is the result of splitting software cost in two; software development cost and software maintenance cost.

	Hardware cost	Develop-ment	Maint. cost	Revenue
Units:	Grade: 1-9	Grade	Grade	Virtual SEK
UF				
1	3,1	3,3	3,4	110 000
2	6,7	6,7	4,6	41 600
3	5,3	3,2	3,4	126 000
4	5,6	6,2	5,1	84 000
5	6,9	7,2	6,3	252 000
6	4,6	5,2	3,4	34 000

Figure 6 Input and output for step 3

This new analysis shows similar efficiency scores:

UF	*Efficiency*	*Efficient*
1	0,98	No
2	0,23	No
3	1,00	Yes
4	0,41	No
5	1,00	Yes
6	0,25	No

Figure 7 Efficiency scores from step 3

All weights of user function 3 and 5 are greater than zero; hence 3 and 5 are regarded as efficient. Little has changed compared to the first analysis. Just like before, virtual revenue (or sales volume) is the dominant factor. Still user function 1 almost 100% efficient.

Concerning the weights, the problems of almost non-zero weights outlined in step 1 are not present in this case, rather the weights are more comparable in size.

5.3.4 STEP 4 –INTRODUCING WEIGHT CONSTRAINTS

The next step of this analysis will be to introduce the two weight constraints previously outlined. So far all evaluation criteria have been assumed equally important. This is however not completely true, something that will be compensated for using weight constraints.

This analysis shows new results:

UF	*Efficiency*	*Efficient*
1	0,94	No
2	0,19	No
3	0,82	No
4	0,41	No
5	1,00	Yes
6	0,21	No

Figure 8 Efficiency scores from step 4

This time only user function 5 complies with the definition of being efficient. The biggest difference compared to the previous step is that user function 3 decreases its efficiency considerably. Before the weight constraints were introduced, the optimization algorithm could compensate for the fact that hardware cost of user function 3 (relative to its other two inputs) was large. Of the analyses conducted so far, this analysis is the most representative of the real world situation, hence it is the most correct.

5.3.5 STEP 5 – SENSITIVITY ANALYSIS

This section investigates what happens if an evaluation criterion is changed.

Sales

In the statistics used, the top 3-segment in sales is functions 5, 3 and 4. The bottom segment is 1, 6 and 2. The three analyses together show the following ranking with respect to efficiency scores of the functions: 5, 3, 4, 1, 6 and 2, which corresponds to an ordering with respect to sales volume. The exact same analysis was then made with the following sales volume:

User Function	*Sales volume*
1	25 000
2	8 000
3	70 000
4	15 000
5	75 000
6	5 000

Figure 9 Result from step 5

The top 3 segment in sales is in this case functions 5, 3, and 1. The bottom segment is 4, 6 and 2. The three analyses together show the following ranking with respect to efficiency scores of the functions: 5, 3, 1, 4, 6 and 2, which also corresponds to an ordering with respect to sales volume. This underlines the importance of sales volume.

Value added

Some examples of the implications of a change in the value added of a user function: If the value added by user function 1 is increased by 15%, then this function becomes efficient, whereas this figure is 25% for user function 3. On the other hand, user functions 2 and 4 need improvements by around 400%, which is unreasonably large. If user function 5 decreases its value added by 10% it is no longer efficient. It should be pointed out that all percentages are interpreted in the cardinal meaning; an increase of a grading with 10% means that it is regarded as 10% more important or 10% more expensive. Moreover, the percentages are approximate values. Since there is only one output, as compared to three inputs, it is clear that a change in revenue is more influential than a change in just one of the inputs.

Hardware cost

A decrease of only 10% would make user function 1 efficient, but a decrease of 35% is required for function 3. If user function 5 increases its hardware cost with 20% it is no longer efficient. For neither user functions 2, 4 and 6 a decrease of 90% is sufficient, which is the same result for development and maintenance costs as well. Even if both of the software related inputs are reduced with 90%, none of them become efficient, however their efficiency scores are increased.

Development cost

For user function 1 a decrease of 20% is required, and a decrease of 80% is necessary for function 3. User function 5 can increase its cost 75% before it is no longer efficient. Comparing these figures to those of hardware cost, it is evident that hardware cost is more influential, which should also be the case considering the weight constraints employed.

Maintenance cost

For user function 1 a decrease of 25% is required, and the corresponding figure for function 3 is 80%. User function 5 can tolerate an increase of 50%. Once again hardware costs prove more influential, but development and maintenance costs are roughly equally important.

5.3.6 ADDITIONAL STEP

Up until now, we have considered the hardware cost only to be incurred when the respective function is chosen. This is overlooking the fact that the ECU must be dimensioned to cope with any customer choice. To reflect this, the hardware cost is multiplied by the sales volume of the corresponding ECU, yielding the true total hardware cost. The revenue is still defined as the product of value added and sales of the user function. It is only in those cases that the customer actually pays for the function.

However, as the inputs are now measured in different units, the two weight constraints above cannot be used. Using no constraint implies that all evaluation factors are regarded as equally important. This is true for the relation between development and maintenance, however not completely accurate for the relations to hardware.

Let us make the following reasonable assumptions:

- User function 1 is chosen in 100% of the cases that the related ECU is mounted.
- User function 2 has its own ECU, that is, the related ECU implements only user function 2.
- User function 3 is chosen in 75% of the cases that the related ECU is mounted.
- User function 4 is chosen in 50% of the cases that the related ECU is mounted.
- User function 5 is implemented by an ECU that is always mounted.
- User function 6 is chosen in 25% of the cases that the related ECU is mounted.

These assumptions in combination with the user function sales presented previously, yields the following ECU sales:

ECU	Sales volume
1	25 000
2	8 000
3	40 000
4	30 000
5	75 000
6	20 000

Figure 10 ECU sales

The corresponding efficiency scores:

	Total hardware cost	Develop-ment	Maint. cost	Revenue
Units:	Grade: 1-9	Grade	Grade	Virtual SEK
UF				
1	3,1	3,3	3,4	110 000
2	6,7	6,7	4,6	41 600
3	5,3	3,2	3,4	126 000
4	5,6	6,2	5,1	84 000
5	6,9	7,2	6,3	252 000
6	4,6	5,2	3,4	34 000

Figure 11 Input and output

UF	Efficiency	Efficient
1	1,00	Yes
2	0,55	No
3	1,00	Yes
4	0,49	No
5	1,00	Yes
6	0,30	No

Figure 12 Efficiency scores

The weights are non-zero, hence user functions 1, 3 and 5 are efficient. The weights of these three are all of the same magnitude, which means than no evaluation criterion is neglected, providing a good overall efficiency measure.

However, having weight constraints would ensure that the overall efficiency is always well reflected.

This incorporation of ECU sales has resulted in an increase in the efficiency scores of functions 2, 4 and 6. User function 5 is still efficient due to strong revenue. However, function 5 has relatively speaking become worse; it is part of an ECU that is always mounted, incurring a very high hardware cost. The worsening of function 5 has made all functions relatively better, increasing their efficiency scores. Most improvement is made by functions 1 and 2, as their portion of the ECU hardware is always used by the function. However, function 1 has in absolute numbers increased very little, as 100% as the maximum score, and the function was already before this final analysis exhibiting a high score.

6 SUMMARY AND CONCLUSION

To address this problem, the following research question was formulated: How can one quantify the resource utilization in embedded systems in the automotive industry?

In order to answer this research question a theoretical framework was created. First of all, the method of Data Envelopment Analysis was explored, and its applicability to this problem setting was explained.

Data Envelopment Analysis can calculate a compound efficiency from the input and output that a particular user function takes in. These inputs and outputs had to be selected and somehow quantified. Based on findings in research papers, the most important factors or evaluation criteria were selected. In order to quantify these evaluation criteria, the main drivers of resource consumption were identified for all of the above mentioned criteria, except for time-to-market and quality. These two factors proved too difficult to estimate with a sufficient accuracy, and had to be omitted in the analysis. By the use of a questionnaire, the performance of a user function regarding the evaluation criteria can be assessed.

The analysis was conducted incrementally, finally providing an analytical model including weight constraints to better reflect the reality. The analysis showed the importance of sales volume and value added, which together form the output revenue. The final step of the analysis was to include sales statistics not only for the user functions, but for the ECUs as well, to better reflect the cost of having to dimension for any customer choice. Taking this into account changed the results of the analysis to a certain extent.

The analysis can be said to confirm the business economics principle that price and sales are decisive factors. For instance, even if a function is very advanced or ingenious it must be sold, and it must be sold at a good price. In conclusion it can be said that the design process is a complex process, and decision support tools may be of great use. It is intuitively appealing to promote design solutions that utilize the available resources in the best way, that is, they are more resource efficient. Hence, the resource utilization framework outlined above may prove very helpful when evaluating historic design decisions, as well as constituting a guideline in current design processes.

7 FUTURE WORK

Some future work remains: the truly interesting part would be to apply the proposed framework on existing user functions, that is, to make an extensive non- fictitious case study. An extension would also incorporate more quantitative data in the analysis. For instance the framework of COCOMO (Constructive Cost Model), created by Barry Boehm [3] may be useful in order to quantify software development costs and software maintenance costs. This would also provide more accurate weight constraints compared to the assumptions made in this work. To include the aspects of time-to-market and quality could also be a future extension to this work.

ACKNOWLEDGMENTS

This work has been financially supported by the Knowledge Foundation and the Swedish Agency for Innovation Systems (VINNOVA).

REFERENCES

[1] Hoch, Detlev et al. : The race to master automotive embedded systems development. . Germany : McKinsey Company, Automotive and assembly sector business technology office, 2006

[2] Merecer Management Consulting and Hypovereinsbank, Studie-Automobiletechnologie 2010. Munchen Aug 2001.

[3] Boehm, B. et.al. Software cost estimation with COCOMO II. Upper Saddle River, New Jersey, USA: Prentice Hall, 2000.

[4] Cooper, W. Data Envelopment Analysis : a comprehensive text with models, applications, references, and DEA-solver software. Hingham, MA, USA: Kluwer Academic Publishers, 1999.

[5] Cooper, W (Editor). Handbook on Data Envelopment Analysis.

[6] Debardelaben, J. et.al. Incorporating cost modeling in embedded-system design. Design and Test of Computers, IEEE, vol. 14, pp. 24-35, July-Sept. 1997.

[7] Fornaciari, W. Cost modeling for embedded digital systems design. (Milan University of Technology, course literature 2006-2007). 2006.

[8] Majumdar, S.K. On the utilization of resources: perspectives from the U.S. telecommunications industry. Strategic Management Journal, Vol. 19, pp 809-831, 1998.

[9] Sommerville, I. Software engineering. Wokingham: Addison-Wesley, 1996.

[10] Sundararaman, K. Design for manufacturability – fault model extensions for RF circuits with an economic perspective for manufacturability. MSc thesis, New York, 2004.

[11] Wennerström, P. En matematisk modell för leverantörsurval. MSc thesis, Stockholm, 2006.
[12] Zientz, W. Electronic systems for commercial vehicles. AutoTechnology 5, pp 40-43, 2007.
[13] Penrose E.T. The theory of the growth of the firm, Oxford university press, New York, 1957

APPENDIX A

EXAMPLE OF QUESTIONNAIRE

Questions regarding user function X (UFx):

1- How many pins does the UFx use?
2 -UFx is advanced.
3 - UFx has an application code that is large.
4 - UFx has an application code that is complex.
5 - UFx requires many calculations.
6 - UFx requires advanced calculations.
7 -UFx uses many variables.
8 - UFx uses many nested or recursive functions.
9 - UFx requires much parameterization.
10 - UFx is distributed/interdependent.
11 - The required reliability of UFx is high.
12 - Re-use of software has not been possible with UFx.
13 - The hardware platform that UFx uses is often upgraded.
14 - UFx is prone to be affected by new regulations (for instance regarding emissions).
15 - UFx is old.
16 - Poor programming style and low quality program documentation have been used.
17 – The wiring harness of UFx is long.
18 - The wiring harness of UFx is located in a harsh environment (e.g. in the engine house).
19 - How necessary is UFx?
20 - How much does UFx increase customer satisfaction?
21 - How does UFx affect the parameter “prestige” of the Product Identity?
22 - How does UFx affect the parameter “performance” of the Product Identity?
23 - How does UFx perform compared to similar user functions among competitors?