SUSTAINABLE EMBEDDED SOFTWARE LIFECYCLE PLANNING

DONG-HYUN LEE¹, KEUN LEE², SOOYONG PARK³, MIKE HINCHNEY⁴, HOH PETER IN¹†

¹Dept. of Computer Science & Engineering, Korea University, Seoul 136-701, Korea
{tellmehenry, hoh_in}@korea.ac.kr

² Semi-Conductor Division, Samsung Electronics Co., Ltd.
Hwasung, Gyunggi-Do 446-711, Korea
gskeun.lee@samsung.com

³Software Engineering Lab., Dept. of Computer Science Sogang University,
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea
sypark@sogang.ac.kr

⁴Lero—The Irish Software Engineering Research Centre, University of Limerick,
Limerick, Ireland
mike.hinchey@lero.ie

Abstract. Time-to-market is a crucial factor in increasing market share in the consumer electronics (CE) market. Furthermore, fierce competition in the market tends to sharply lower the prices of brand-new CE products as soon as they are released. Software-intensive embedded system design methods such as hardware/software co-design have been studied with the goal of reducing development lead-time by designing hardware and software simultaneously. Many researchers, however, concentrate on static design methods—in which design remains unchanged once determined. To survive this deadly market competition, a dynamic design strategy that takes various market conditions into account is needed for software-intensive embedded systems. In this paper, a sustainable embedded software lifecycle planning (SESLIP) process based on the evolution of embedded software is proposed. The SESLIP process provides a dynamic method for both selecting product lifecycle design alternatives and generating a profit-maximizing transition plan that covers the entire lifecycle of a product.

Keywords: Software Engineering, Product lifecycle, Embedded systems, Embedded software evolution

1 Introduction

Consumer electronics (CE) markets are so competitive that the price of a CE product rapidly decreases after its release [1]. According to a recent WitsView TV market report [2], the price of 42” LED TVs fell from $1,229 in September 2010 to $708 in September 2011—a drastic fall in price of 42.4% in one year. Thus, CE companies strive to reduce production cost before release of products in order to survive in the market. However, changing market conditions after release lead to reduced production cost not only before release, but also throughout the lifecycle of a product. This means that in order to maintain price competitiveness a product lifecycle plan is essential for CE companies. What kind of product lifecycle plan can be used to sustain the profit on CE products whose prices tend to fall sharply in this cut-throat competitive market?

† Hoh Peter In is the corresponding author.
Our methodology for planning the product lifecycle provides a way to find an adaptive transition path for hardware/software design alternatives that takes market conditions into consideration in the early stages of the development phase. The key driving factor is the replacement of hardware components with software alternatives [3, 4, 5, 6]. For example, a software-defined radio can be used to replace the functions of the radio frequency chips in cellphones. This means that it is possible to have various candidates for the radio component. Production costs are dramatically reduced by replacing hardware chips with software programs. However, the size of the software increases as more functions are shifted from hardware to software. Rapid growth in software size leads to an elongated development lead time and higher costs, since software complexity and labor cost are also rising. Consequently, using different development lead times and lower cost software alternatives, it is possible to generate production plans that have multiple design options and various combinations of hardware and software; resulting in reduced production cost from time-to-market to the end of the lifecycle.

We propose a sustainable embedded software lifecycle planning (SESLIP) process for generating transition sequences for product design alternatives enabled by shifting from hardware to software, in order to maximize profits by reducing production cost. The proposed SESLIP process is performed in the early phase of development for an adaptive production plan that maximizes profits.

Even though research has already been done on the impact of cost on embedded systems design [1, 7, 8], there is a major difference between what has been done and our proposed methodology. The approaches taken in existing research are static: once the product design is decided on in the development phase, the design is not changed after release. In contrast, our proposed SESLIP process is adaptive: by spanning the product design to the end of the lifecycle, the production plan allows modification of the product design for profit maximization. The result of the SESLIP process is a reference production plan that can be used by CE vendors to manage production and adaptively adjust to market conditions.

2 Related Work

Existing research done on software-intensive embedded systems have mainly focused on developing architectures, platforms, and development environments for embedded systems—especially various resource utilization aspects such as processor, memory, size, power consumption, and timing [9]. The majority of traditional research endeavors, however, do not take into account factors such as marketing conditions and business requirements.

Very few studies have ever attempted to examine the cost impacts of a software-intensive embedded system design decision. Funded by the US Defense Advanced Research Projects Agency’s Rapid Prototyping of Application Specific Signal Processors (RASSP) program, Debadelaben et al. [7] proposed a pioneering approach for improving embedded systems design processes by incorporating cost models. They adopted the Revised Intermediate COCOMO (REVIC) [10, 11] model to estimate software development costs and determined hardware costs by summation of commercial off-the-shelf (COTS) hardware components.

Ragan et al. [8] proposed a cost model for the system-on-chip design process and a tool called Ghost for trade-off analysis with hardware and software components in a chip. Ghost helps to choose optimum hardware/software partitioning tailored for both the wafer fabricator and the software development. The Ghost model also utilizes the COCOMO model for estimating the costs of developing software. They also proposed a hardware development cost modeling process that is based on the number of gates on a chip.

Lee et al. [1] proposed a hardware/software co-design process that works by progressively modifying hardware/software design options throughout the product lifecycle. This proposed process also includes a lifecycle process for cost reduction, but it is focused on hardware/software co-design, not embedded software evolution. Furthermore, the lifecycle process is not concrete due to the lack of detailed processes.

Previous work is considered pioneering because the cost factor is considered as an essential attribute in the software-intensive embedded system design in spite of the focus on a narrow area of the system-on-a-chip. However, the objective of previous research was to select the best design among various design options. Once the design is
determined then it is not changed after release. Even though market conditions change significantly after a product’s release, it is hard to adapt to market conditions due to the lack of a production plan for the product. Therefore, in the partitioning stage of software-intensive embedded systems development, an adaptive transition plan of design alternatives is needed to flexibly cope with market condition changes throughout the entire product lifecycle.

3 A Sustainable Embedded Software Lifecycle Planning (SESLIP) Process

The objective of the SESLIP process is to provide a lifecycle plan that enables software-intensive embedded systems to adapt to market conditions. The SESLIP process (depicted in Figure 1) defines an adaptive transition sequence plan through the lifecycle. The process comprises four steps, which we will explain using a robot cleaner example, as depicted in Figure 1. We obtained development information (Table 1) for the robot cleaner from a global electronics company. The development data have been abstracted due to the confidentiality policy of the company and, for this reason also, the marketing data used in Step 4 and the next section is hypothetical.

![Figure 1. Overview of the complete SESLIP process](image)

- **Step 1: Identification of components and candidates**

  In embedded systems development processes, the functions of a CE product are typically assigned to hardware or software components in the partitioning phase. Hardware and software developers perform partitioning decisions with system requirements in order to achieve the desired functionality of the product. During the partitioning process, the decision as to whether to implement certain functions as hardware or software is usually controversial. Finally, components are specified as software-intensive, hardware-intensive, and hardware or software enabled components, then candidate components are identified: software-intensive and hardware-intensive components have one implementation candidate, respectively, while hardware or software enabled components have multiple candidates.

  Step 1 in Figure 1 shows a robot cleaner that has four components that facilitate movement: control (MC), noise control (NC), battery management (BM), and cleaning algorithm (CA). MC is primarily a hardware-intensive
component that enables the robot cleaner to move, whereas CA is a software-intensive component that enables the robot cleaner to navigate in a smart fashion. The remaining two components, NC and BM, which have two candidates respectively, allow for design alternatives. After finding all candidate components, development cost, development lead time, and the production cost for each candidate are analyzed. Table 1(a) gives detailed information on the robot cleaner. The development cost for HW is the total bill-of-material (BOM) cost in a candidate component and the company estimated SW development cost by labor cost of SW engineers for a candidate. To simplify the process, we assume that candidates for a component, i.e., NC₁ and NC₂, have the same performance.

- **Step 2: Generation of product design alternatives using candidate components**
  The second step of the SESLIP process aims to generate product design alternatives by combining candidate components. Step 2 in Figure 1 shows that the robot cleaner has four design alternatives, A₁, A₂, A₃, and A₄. If we assume that all the candidate components start to develop at the same time, the development lead time of a design alternative is in accordance with the latest development lead time of component candidates in the alternative. In the case of alternative A₃, the development lead time is 12 months due to the development lead time of BM₂. The production cost of each alternative is assessed based on the candidate components. For example, the production cost of A₂ is the sum of the production costs of MC₁, NC₂, BM₁, and CA₁. Table 1(b) summarizes the development lead time, development cost, and the production cost of each product design alternative derived from Table 1(a).

<table>
<thead>
<tr>
<th>Component</th>
<th>Component candidate</th>
<th>Development Cost</th>
<th>Development lead time (months)</th>
<th>Production Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movement Control (MC)</td>
<td>MC₁</td>
<td>10,000</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>NC₁</td>
<td>12,000</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>NC₂</td>
<td>7,000</td>
<td>7</td>
<td>40</td>
</tr>
<tr>
<td>Noise Control (NC)</td>
<td>BM₁</td>
<td>6,000</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>BM₂</td>
<td>6,000</td>
<td>12</td>
<td>60</td>
</tr>
<tr>
<td>Battery Management (BM)</td>
<td>CA₁</td>
<td>2,333</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Cleaning Algorithm (CA)</td>
<td>CA₁</td>
<td>2,333</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design alternative</th>
<th>Component candidates</th>
<th>Development lead time of design alternative (months)</th>
<th>Production cost of design alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₁</td>
<td>MC₁, NC₁, BM₁, CA₁</td>
<td>6</td>
<td>180</td>
</tr>
<tr>
<td>A₂</td>
<td>MC₁, NC₂, BM₁, CA₁</td>
<td>7</td>
<td>170</td>
</tr>
<tr>
<td>A₃</td>
<td>MC₁, NC₁, BM₂, CA₁</td>
<td>12</td>
<td>160</td>
</tr>
<tr>
<td>A₄</td>
<td>MC₁, NC₂, BM₂, CA₁</td>
<td>12</td>
<td>150</td>
</tr>
</tbody>
</table>

- **Step 3: Product design alternatives transition planning**
  With the product design alternatives from step 2, we generate all the possible transition plans for the design alternatives depicted in Figure 1. This results in 7 such plans for the robot cleaner. In Figure 2, R signifies the end of development and the time at which alternative A₁ (the first version of the robot cleaner) is immediately released. Plans 1 to 7 are possible alternative plans generated from A₁ at R. Plan 1 is a fixed plan with A₁, while Plans 2 to 7 adaptively change from A₁ to A₄. Here, a path A₁⇒A₂⇒A₃ is not considered because the key driver of our proposed process is the lowering of production cost by changing from hardware to software (as stated in Section 1). Among all the plans, which plan maximizes profit? In our next step, we perform economic analysis in order to identify this plan.
  The development cost of each plan depends on the design alternatives in the plan and the development lead time.
of the candidate components. For example, Plan 3 in Figure 1 has alternatives $A_1$ and $A_1 (MC_1, NC_1, BM_2, CA_1)$. Consequently, $MC_1, NC_1, BM_2,$ and $CA_1$ should be developed for Plan 3. $MC_1, NC_1, BM_1,$ and $CA_1$ (for $A_1$) are completed within time $R$, and $BM_2$ is complete at time $R+6$ for $A_2$. Let $g_s(t)$ be the development cost of Plan $s$ at time $t$, and $T$ be the end of the lifecycle of a robot cleaner, then $g_s(t)$ for Plan 3 is as follows:

$$g_3(t) = \begin{cases} 
\text{the sum of development cost } MC_1, NC_1, BM_1, \text{ and } CA_1 = 94333, & R \leq t < R + 6 \\
\text{the sum of development cost } MC_1, NC_1, BM_1, BM_2, \text{ and } CA_1 = 132333, & R + 6 \leq t \leq T 
\end{cases}$$

If the development cost of Plan $s$ at time $t_k > 0$, that is, $g_s(t_k)$ is over the budget, Plan $s$ is discarded. Here, we assume that all the plans (Plans 1 to 7) are within budget. Step 3 in Figure 1 shows the development costs of the other plans. The production cost for Plan $s$ is also similar to the development cost. Considering the same scenario as Plan 4, let $f_s(t)$ be the production cost of Plan $s$, then $f_s(t)$ for Plan 4 is as follows:

$$f_3(t) = \begin{cases} 
\text{production cost of } A_1 = 180, & R \leq t < R + 6 \\
\text{production cost of } A_3 = 160, & R + 6 \leq t \leq T 
\end{cases}$$

**Step 4: Transition plan selection**

The goal of the SESLIP process is to determine the transition product design alternative that maximizes profit. For economic analysis of the plans in Step 3, prediction of sales volume and price trend is required in order to calculate the profit. Let $T$ be the end of the lifecycle of the robot cleaner. Then the problem now boils down to finding a plan $s$ that maximizes profit $\pi$. To simplify the profit model, we exclude discount rates according to the assumption of short lifecycle in the CE market, and we consider production cost to be unit variable cost and development cost as fixed cost.

$$\max_{Plan_s} \pi = \text{Units Sold} \times (\text{Unit sales price} - \text{Unit variable cost}) - \text{Fixed cost} = \int_{R}^{T} (p(t) - f_s(t)) \cdot x(t) \, dt - g_s(T) \tag{1}$$

Where:

1. $p(t)$ is price prediction. In a competitive market, companies are price takers, while companies are able to determine pricing in a monopolistic market. $p(t)$ depends on markets and companies.
2. $f_s(t)$ is the production cost function of Plan $s$.
3. $x(t)$ is the sales rate at $t$. Let $X(t)$ be cumulative sales volume, then $x(t) = dX(t)/dt$.
4. $g_s(t)$ is the development cost of Plan $s$.

Figure 2 shows the arithmetically obtained profit for each plan using formula (1) throughout the lifecycle of the robot cleaner. (Due to the company’s confidentiality clause, we were not allowed to use actual market data. As a result, we carried out simulations using hypothetical market data.) Given lifecycle $L = 12$ months, a constant price $p(t) = 250$ USD, and sales rate (sales volume per month) as in Figure 2(b), Plan 7 ($A_1 \rightarrow A_2 \rightarrow A_3$) maximizes cumulative profit during the lifecycle of the robot cleaner, which is 25.4% more than the cumulative profit of the fixed plan (Plan 1) at the end of the lifecycle.
Figure 2. Profit analysis for selecting transition plan

4 Discussion

4.1 Sensitivity of profit analysis

As shown in formula (1), the unit sales price $p(t)$, unit production cost $f_s(t)$, sales volume $X(t)$, and development cost $g_s(t)$ determine the profit. Here, we consider two factors associated with sensitivity of the profit analysis: price and time-to-market.

- **Price sensitivity**
  Price plummets when there is severe competition among vendors in a cut-throat market. We can easily surmise that the more prices fall, the less profit there is. Here, we consider various rates at which prices plummet for a scenario in which prices fall almost 60% at the end of the lifecycle, compared with the initial price of the robot.
cleaner, as depicted in Figure 3(a). Price 1 is the fixed price considered in Figure 2. Price 2 shows a positive plummeting rate, which means that the price falls slowly early in the lifecycle and falls sharply later on in the lifecycle. Conversely, the plummeting rates of prices 3 and 4 are less than that of price 2. Figures 3(b), 3(c), and 3(d) depict the cumulative profit at prices 2, 3, and 4, respectively. The cumulative profit at price 1 is shown in Figure 2(d). Given the sales volume depicted in Figure 2(b), the result of profit analysis at each price plunge shows that Plan 7 is still the best transition plan since Plan 7 continuously lowers unit production cost. In the case of price 2, Plan 7 results in approximately 43% more profit than the fixed plan. Moreover, in Figure 3(c), Plan 7 is the only plan with a positive profit during the lifecycle of the robot cleaner. Therefore, using adaptive planning for the robot cleaner it is possible to have positive cumulative profit around the price plummeting rate of price 5. If price is predicted as price 4 in Figure 3(d), vendors are able to decide whether to stop development of the robot cleaner because all the plans at price 4 are below zero.

![Figure 3. Price sensitivity for profit analysis](image-url)
Time-to-market sensitivity

Bringing forward a product’s time-to-market is critical in competitive markets. Debardelaben et al. [7] formulated a relation between delayed time-to-market and revenue loss as follows: \( r_L = R_0 d/(3Wd) \) where \( r_L \) is revenue loss, \( R_0 \) is expected product revenue, \( d \) is delayed time-to-market, and \( W \) is half of the lifecycle. We will now consider what happens when all the plans are delayed for 2 months, as in the two scenarios shown in Figure 4(c): constant price (price 1) and price plummeting at a constant rate (price 2). We obtained the sales rate for a 2-month delay in Figure 4(b) by applying the revenue loss equation. In the case of price 1, Figure 2(d) shows that the cumulative profit of Plan 7 without delay is 25.4% more than that of the fixed plan, however, delayed time-to-market causes the cumulative profit of Plan 7 to plummet to 71.2% that of the fixed plan with no delay, as illustrated in Figure 4(d). However, in the case of price 2, the cumulative profit of Plan 7 outpaces that of the fixed plan with no delay after \( R+8 \) despite the 2-month delayed time-to-market, and the cumulative profits of Plan 4, Plan 6, and Plan 7 outpace the fixed plan with no delay at the end of the lifecycle (Figure 4(e)). This means that adaptive planning is capable of compensating for profit loss of short duration delayed time-to-market in markets where prices are falling.
4.2 Threats to validity

There are some threats to the validity of this study resulting from the underlying assumptions elaborated on below.

- **The SESLIP process is not applicable to every embedded systems industry**: Our motivation is to reflect the economic value of replacing hardware with software \[6\] in order to reduce production cost during the product lifecycle. Thus, this study has significance for massive embedded systems industries, such as CE industries, in which profit maximization through cost reduction is the main priority during the product lifecycle for surviving in a competitive market. In contrast, safety or dependability is the main priority during the lifecycle of medical embedded systems such as surgery robots that have a direct influence on human life while in operation.

- **Not all embedded systems are under a component-based development environment**: We assumed a product with exclusive components that have no effect on modification. However, in reality, some components are composed of tightly coupled hardware and software. In such a scenario, the possibility exists that transition from hardware to software causes overall architectural changes due to lack of flexibility in design, resulting in overhead costs during the transition of the design alternatives.

- **Performance of software components is not equal to that of corresponding hardware components**: Unlike our assumption of performance equivalence between hardware components and corresponding software, hardware performance is, in general, better than software programs. Consequently, some candidates may fail to operate as expected. An acceptance criteria or threshold analysis is needed in the transition of design alternatives.

5 Conclusion

Many software-intensive embedded system design corporations are endeavoring to reduce production costs, and to shorten their development schedules. Research has been, and is still being, conducted to develop new design methodologies for software-intensive embedded systems in order to resolve this issue. A considerable amount of research has focused only on technical aspects; namely, how to partition hardware and software components, or what process is suitable for developing an embedded system. However, the ever-changing conditions in the market
have forced companies in the industry to amend their designs to cope with flexible market conditions such as time-
to-market pressures and plummeting product prices.

In this paper, we proposed a sustainable embedded software lifecycle planning (SESLIP) process for maximizing
profit in severely competitive markets by adaptive lifecycle planning of changing hardware with software. We
considered not only development cost and lead-time, but also production cost and product lifecycle, based on the
characteristics of embedded software development. Other research on embedded systems focused on technical
aspects of the development phase without considering production phase. Consequently, in practice, we believe
SESLIP will be much more helpful to software-intensive embedded systems companies than existing design
methods for software-intensive embedded systems.

For future work, firstly, research on acceptance criteria or threshold analysis is needed for the replacement of
hardware with software, as mentioned above. Secondly, the profit analysis model needs to be improved. We
assumed that unit production cost is the only factor for unit variable cost, but overhead cost occurs when design
alternatives change. Software maintenance cost also needs to be taken into consideration as software size increases.

6 Acknowledgements

This research was supported by the Ministry of Knowledge Economy (MKE), Korea, under the Information
Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency
(NIPA-2011-C1090-1131-0008). This research was also supported under the program for Software Engineering
Technologies Development by NIPA, and in part by a Science Foundation Ireland grant 10/CE/11855 to Lero—the
Irish Software Engineering Research Centre (www.lero.ie). For further information, contact the corresponding
author, Hoh P. In at hoh_in@korea.ac.kr.

References

Codesign Life-Cycle Model”, IEEE Computer, Volume 42, Issue 2, 2009, Page(s): 100 – 102
BETWEEN CONNECTED- ELECTRONIC DEVICES, Springer Netherlands, 2005, Page(s): 139-165
Chapter 14 from SOFTWARE PRODUCT LINES IN ACTION, Springer Berlin Heidelberg, 2007, Page(s): 219-232
[8] DANIEL RAGAN, PETER SANDBORN, PAUL STOAKS, “A Detailed Cost Model for Concurrent Use With Hardware Software co-