Chapter 3
Evolutionary Architecting of Embedded and Enterprise Software and Systems

Jakob Axelsson
Swedish Institute of Computer Science (SICS), Sweden & Mälardalen University, Sweden

ABSTRACT

Many industries rely heavily on embedded software and systems to maximize business value in their products. These systems are very complex, and the architecture is important to control the complexity and make development efficient. There are often also connections between the embedded system and the different lifecycle processes, and hence, to the enterprise systems supporting those processes. It is rare to start from scratch when developing new products, and instead, these companies evolve their products over time, which means that architecting needs to be evolutionary. This chapter describes what such an evolutionary architecting process can look like based on observations from industry, and how the process can be continuously improved using a maturity model. It is also presented how the embedded system relates to different elements of the enterprise architecture.

INTRODUCTION

In many companies developing technical products, such as the automotive industry, process automation, defense, or telecommunication, embedded systems and software play an increasingly important role. The embedded systems have developed from simple, stand-alone processors into a large number of computers with distribution networks and millions of lines of software. Each processor node can have many actuators and sensors, which are sometimes very information intensive. There are also often links (wired or wireless) to other systems for information exchange. The embedded system needs to be packaged in whatever space is available in the product, and be equipped with an adequate power supply system, which often introduces both new operating modes, and electrical requirements that have implication on the software in different ways. (For an overview

DOI: 10.4018/978-1-4666-2199-2.ch003

of contemporary complex embedded systems and their challenges, see e.g. Broy *et al.*, 2007; Grimm 2003).

This increasing complexity leads to soaring development costs, and many companies strive to curb this trend by reusing software and hardware between products. At the same time, the development cost targets have to be balanced against the strict requirements of these products. They are often business-critical, or even safety-critical; they are installed in harsh environments and must be very robust against disturbances; they have a long lifetime where it is usually not acceptable to discard the overall product just because of a failure in the embedded system; and finally, they often have to be flexible to adapt to changes and upgrades after delivery to customers. This means that a substantial part of the design effort deals with supporting other lifecycle stages than just normal operation.

With these requirements, and a multiplicity of products and variants for different customers and markets, the architecture is becoming very important and is a source of increasing interest for companies developing embedded systems. An architecture can be defined as "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" (IEEE, 2000). Often, a product line approach is applied, where the same platform is used as a basis, with modifications to fit individual products and customers. The decisions made by architects in the early phases influence many decisions made later on, and the architecting decisions are difficult to change further down the process (Smith and Reinertsen, 1995). With a poor architecture, downstream development activities will thus become much more expensive and time consuming.

In the embedded systems literature, there is a strong focus on systems and hardware architecture (see e.g. Sangiovanni-Vincentelli and Di Natale 2007), and this is natural since custom hardware is almost always used and there are hence many degrees of design freedom in the choice of hardware components and structure. Over the years, this focus has been complemented with increasing emphasis on software architecture as a consequence of the rising importance of software in these products. However, what is often neglected in the literature is a discussion on the links between the embedded systems in the product and the enterprise architecture, i.e. the organizing logic and IT infrastructure, of the firms that produce it or interact with it over its lifetime.

We have previously done in-depth studies of the current architecting practices at a few automotive companies (Wallin and Axelsson, 2008; Wallin *et al.* 2009). The issues found were later validated also in other industrial areas (Wallin *et al.* 2012) where embedded software and systems play an essential role. Among the issues were a lack of processes for architecture development, and that the organizations had an unclear responsibility for architectural issues. Also, there was a lack of long-term strategy to ensure that legacy does not negatively impact future decisions, and a lack of methods to evaluate the business value when choosing the architecture. In short, the organizations rely on the performance and knowledge of individuals instead of on processes and methods (a finding also supported by Hoorn *et al.*, 2011; Babar and Gorton, 2009). Several of the issues also directly relate to the fact that the embedded systems architecture is linked to enterprise architectures. This includes trade-offs when the same platform is used by several firms; relations to suppliers; and decision making which is often narrow in scope to the embedded system (or parts of it). It is indicated that the complexity of the organization and product has grown beyond what the current processes can handle.

Based on this information collected from industry, it appears plausible to assume that a mature organization would work with architecting of embedded systems and software mainly through stepwise refinement rather than large leaps. We