Chapter II

Challenges in Requirements Engineering for Embedded Systems

Eman Nasr, Cairo University, Egypt

Abstract

In this chapter we are particularly interested in requirements engineering of software where the software is part of a complex engineered system; that is, embedded software. Embedded software is the software that runs on a computer system that is integral to a larger system whose primary purpose is not computational. Embedded software systems are of rising significance in the industry, and they are found in a wide range of industries in the modern world, including medical, nuclear, chemical, rail networks, aerospace, and automotive industries. The RE of this category of software is challenging because of its special properties that add to its complexity and make it more expensive and error-prone as compared with other software categories, for example, business applications. In this chapter we identify the special properties of embedded software systems to help in better understanding of such domain, discuss the special RE challenges that the special properties introduce, and introduce the main current RE approaches for the domain.
Introduction

Modern computer-based systems are becoming increasingly complex ensembles of hardware and software, thus adding more challenges to the software requirements engineering process. Requirements engineering (RE) is usually known as the branch of software engineering that deals with the early phase of software development. Although there is no single definition for RE, because the field of research is still maturing, a well-accepted definition is (Zave, 1995):

"Requirements engineering is the branch of software engineering concerned with the real-world goals for functions of, and constraints on, software systems."

RE deals with activities that attempt to understand the exact needs for a software-intensive system and to translate such needs into unambiguous requirements that will be used in the development and testing of the software system. RE is considered a combination of mainly three interacting activities: eliciting requirements related to a problem domain, specifying the requirements in an unambiguous way, and ensuring the validity of such requirements (Loucopoulos & Karakostas, 1995). It is one of the most important activities in software development because errors made in requirements become increasingly costly to repair later in development and extremely costly to repair after delivery (Brooks, 1987; Heitmeyer, 1997; Hofmann, 1993; Wieringa, 1996). The product of RE is a requirements specification, which forms a foundation for the whole subsequent/concurrent development. Among the important properties of a good requirements specification are: completeness, lack of ambiguity, good structure, and ease of understanding by all of the stakeholders involved in the software system. A good requirements specification should seek to bridge the communication gap between domain experts and software experts. It is widely accepted that a good RE method is crucial for any successful large-scale software system development. It is also widely recognised that the most serious embedded software failures can be traced back to defective specification of requirements (Knight, 2002; Leveson, 1995; Lutz, 1993).

In this chapter we are particularly interested in RE of software where the software is part of a complex engineered system, that is, embedded software. Embedded software is the software that runs on a computer system that is integral to a larger system whose primary purpose is not computational (Lutz, 1993). An embedded software system usually provides at least partial control over the hardware system in which it is embedded. An embedded software system is usually highly reactive, as it responds to various sensor inputs, interrupts, or alarm conditions from its environment. Embedded software systems are of rising significance in the industry, and they are found in a wide range of industries in the modern world, including medical, nuclear, chemical, rail networks, aerospace, and automotive industries. Embedded software systems have proliferated almost everywhere over the past few years, from household appliances, like toasters and washing machines, to cars to aircraft and spacecraft. Of course the software embedded in these products varies in complexity as widely as the style of the products.
The Development of an Optimised Metadata Application Profile
www.igi-global.com/chapter/the-development-of-an-optimised-metadata-application-profile/188231?camid=4v1a