DERAF: A High-Level Aspects Framework for Distributed Embedded Real-Time Systems Design

Edison Pignaton de Freitas[1], Marco Aurélio Wehrmeister[1], Elias Teodoro Silva Jr.[1], Fabiano Costa Carvalho[1], Carlos Eduardo Pereira[1,2], and Flávio Rech Wagner[1]

[1] Computer Science Institute – Federal University of Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
[2] Electrical Engineering Department, Federal University of Rio Grande do Sul, Brazil
{epfreitas, mawehrmeister, etsilvajr, fccarvalho, flavio}@inf.ufrgs.br, cpereira@ece.ufrgs.br

Abstract. Distributed Embedded Real-time Systems (DERTS) have several requirements directly related to characteristics that are difficult to handle when a pure object-oriented method is used for their development. These requirements are called Non-Functional Requirements (NFR) and refer to orthogonal properties, conditions, and restrictions that are spread out over the system. Pure object-oriented methods do not address successfully those concerns, so new technologies, like aspect orientation, are being applied in order to fulfill this gap. This work presents a proposal to use aspect orientation in the analysis and design of DERTS. To support our proposal, we created DERAF (Distributed Embedded Real-time Aspects Framework), an extensible high-level framework (i.e. implementation-independent) to handle NFR of DERTS. DERAF is used together with RT-UML in the design phase, aiming to separate the handling of non-functional from functional requirements in the Model Driven Design of DERTS. A qualitative assessment of DERAF separation of concerns is also presented.

Keywords: Requirements Specification, Distributed Real-time Embedded Systems, Aspect-Orientation applied to DERTS, Separation of DERT Concerns.

1 Introduction

The increasing complexity of Distributed Embedded Real-Time Systems (DERTS) requires new development techniques in order to support system evolution and maintainability, and the reuse of previously developed artifacts. An important concern involved in DERTS design is how to deal with Non-Functional Requirements (NFR), which have crosscutting concerns, that means, some NFRs may affect very distinct parts of the system under development. If not properly handled, NFRs are responsible for tangled code and loss of cohesion. In the literature, it is possible to find several references addressing this separation of concerns where the crosscutting concerns are identified as NFRs, as in [1], [4] and [6]. In order to promote a separation of concerns, guidelines to handle NFRs separately from the functional ones have been proposed, using concepts such as subject-oriented programming [2] and aspect-oriented

A. Moreira and J. Grundy (Eds.): Early Aspects 2007 Workshop, LNCS 4765, pp. 55 – 74, 2007.

programming [3]. Both approaches address the problem at the implementation level. Some other approaches propose to take NFR into account as soon as possible that means, in the early phases of the specification and design, as in the Early-Aspects [4] approach.

Real-time systems have a very important NFR which is the concern about timing aspects, such as deadlines, maximum jitter, worst case execution time, tolerated delays, and other. The complexity related to the non-functional analysis of these systems increases when they become distributed and embedded. To deal with some of these NFR, some proposals suggest the use of aspects, as in [5] and [6] Aspects can help to deal with crosscutting NFRs into DERTS design, modularizing their handling. Besides the modularization capability, more abstract aspects are easier to reuse because they define the handling of a concern at high-level and also how it can be applied (or how it affects) the system without implementation or platform constraints.

This work presents the Distributed Embedded Real-time Aspects Framework (DERAF), which provides an extensible set of aspects to deal with NFRs of DERTS at a high-level of abstraction. On other words, DERAF is a set of implementation-independent aspects to handle NFRs during the creation of DERTS RT-UML [10][11] models at design phase. In this initial version, DERAF handles the following NFRs (see Section 2): timing, precision, performance, distribution, and embedded behavior. DERAF was created to be an extensible framework, such that it can be extended to include support to other important aspects (e.g. fault-tolerance). However, these new aspects must follow the high-level nature of the framework and also the implementation-independence. It is important to highlight that, in spite of the high-level of abstraction, aspects within DERAF must be implementable. On other words, the realization of each aspect must be possible, avoiding unfeasible aspects.

The remaining of this paper is organized as follows. Section 2 outlines a brief discussion on NFRs within DERTS domain. The following section presents an aspects framework to handle the identified NFRs. Section 4 presents the transition from requirements to design of DERTS using the DERAF. A case study and a qualitative assessment of final design are presented in Section 5. Finally, the related work is presented in section 6, and final remarks and future work in Section 7.

2 DERTS Non-functional Requirements

In order to deal with NFRs in early design phases of DERTS, it is firstly necessary to define and understand the main concepts involved in the system context. The need of this information motivates the classification of these concepts in terms of non-functional concerns, which can affect the behavior and structure of the DERTS being designed. Fig. 1 presents some key requirements related to DERTS development, which are mainly based on the study presented in [14], on the IEEE glossary [15] and on the SEI glossary [16].

The real-time concern is captured by the requirements stated in the *Time* classification, which is divided in *Timing* and *Precision* requirements. The first one is concerned with the specification of temporal limits for system activities execution, such as established deadlines and periodic activations. Requirements classified as