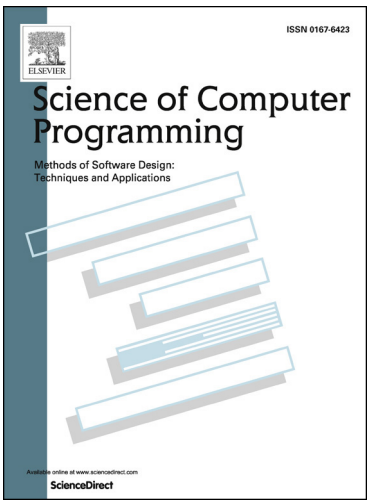
Accepted Manuscript



Towards the systematic analysis of non-functional properties in model-based engineering for real-time embedded systems

Guillaume Brau, Jérôme Hugues, Nicolas Navet

PII: S0167-6423(17)30292-7
DOI: https://doi.org/10.1016/j.scico.2017.12.007
Reference: SCICO 2177

To appear in: *Science of Computer Programming*

Received date: 16 June 2016
Revised date: 6 December 2017
Accepted date: 18 December 2017

Please cite this article in press as: G. Brau et al., Towards the systematic analysis of non-functional properties in model-based engineering for real-time embedded systems, *Sci. Comput. Program.* (2017), https://doi.org/10.1016/j.scico.2017.12.007

Towards the Systematic Analysis of Non-Functional Properties in Model-Based Engineering for Real-Time Embedded Systems

Guillaume Brau[a,b], Jérôme Hugues[b], Nicolas Navet[a]

[a] *University of Luxembourg, CSC Research Unit,*
6 rue R. Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg.
{guillaume.brau, nicolas.navet}@uni.lu
[b] *Université Fédérale Toulouse Midi-Pyrénées, ISAE-SUPAERO,*
10 avenue E. Belin, 31055 Toulouse, France.
jerome.hugues@isae-supaero.fr

Abstract

The real-time scheduling theory provides analytical methods to assess the temporal predictability of embedded systems. Nevertheless, their use is limited in a Model-Based Systems Engineering approach. In fact, the large number of applicability conditions makes the use of real-time scheduling analysis tedious and error-prone. Key issues are left to the engineers: *when to apply a real-time scheduling analysis? What to do with the analysis results?* This article presents an approach to systematize and then automate the analysis of non-functional properties in Model-Based Systems Engineering. First, *preconditions* and *postconditions* define the applicability of an analysis. In addition, *contracts* specify the analysis interfaces, thereby enabling to reason about the analysis process. We present a proof-of-concept implementation of our approach using a combination of constraint languages (REAL for run-time analysis) and specification languages (Alloy for describing interfaces and reasoning about them). This approach is experimented on architectural models written with the Architecture Analysis and Design Language (AADL).

Keywords: Model-Based Systems Engineering; Non-Functional Properties; Analysis; Contracts; Real-Time Scheduling; Architecture Description Languages

1. Introduction

Context. Embedded systems have become an integral part of our daily life. We can found them in cars, aircrafts, trains, robots, healthcare equipments, mobile phones, consumer electronics, etc. In particular, a major issue related to embedded systems is to fulfill the non-functional requirements dictated by their environment, expressed for example in terms of timing, dependability, security, or other performance criteria. In safety-critical applications for instance (e.g. in an airplane), missing a non-functional requirement can have severe consequences, e.g. loss of life, personal injury, equipment damage, environmental disaster, etc.