Available online at www.sciencedirect.com
SCIENCE DIRECT
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 116 (2005) 185–198
www.elsevier.com/locate/entcs

A Compositional Framework for Formally Verifying Modular Systems[1]

Carlo A. Furia and Matteo Rossi[2]

Dipartimento di Elettronica e Informazione, Politecnico di Milano
32, Piazza Leonardo da Vinci, 20133 Milano, Italy

Abstract

We present a tool-supported framework for proving that the composition of the behaviors of the separate parts of a complex system ensures a desired global property of the overall system. A compositional inference rule is formally introduced and encoded in the logic of the PVS theorem prover. Methodological considerations on the usage of the inference rule are presented, and the framework is then used to prove a meaningful property of a simple, but significant, control system.

Keywords: Formal verification, modular systems, real-time, compositionality.

1 Introduction

As systems grow in size, being able to subdivide them in components is of crucial importance to keep their complexity under control. In particular, one would like to specify and design single components separately, and then be able to guarantee that they also behave correctly when they interact with each other. In fact, parts that operate properly under some assumptions on the behavior of the external world might misbehave when interacting with other elements of the overall system that do not satisfy those assumptions (for example, a data analyzer that accepts inputs with frequency r might

[1] Work supported by the MIUR project: "QUACK: Piattaforma per la qualità di sistemi embedded integrati di nuova generazione"
[2] Email: rossi@elet.polimi.it

doi:10.1016/j.entcs.2004.02.076

work improperly when connected to a sensor that sends data with frequency $2r$).

Formal methods are more and more recognized to be a useful tool for the development of applications, especially critical ones, as they allow to precisely verify the correctness of systems in their early development phases, before uncaught mistakes become overly costly to fix, or even catastrophic. One problem often attributed to formal methods, however, is that they do not "scale up", i.e. when the system grows in complexity, they are too cumbersome and unwieldy to be used effectively.

A compositional framework can help in this regard, in that it would allow one to focus on the single parts of the system at first, and analyze their mutual interactions at a later moment, with a smaller effort than it would be required if all aspects (local and global) of the application were taken into account at once at integration time. A good, proof-oriented compositional framework must be based on sound inference rules that allow one to deduce global properties of the system from the behavior of its single parts. In addition, for the framework to be actually usable, it should be supported by (semi)automatic tools that facilitate the analysis of the modeled systems.

Rules for composing single specifications into complex systems have been studied in the past [2], also, but not only, with reference to temporal logics [1] [3]. This paper presents a valid inference rule for the TRIO specification language [5,6] that is suitable to formally prove the correctness of the behavior of a modular system from the behavior of its components. The rule has been encoded in the logic of the PVS theorem prover [7], and support strategies have been developed.

The paper is structured as follows: Section 2 shortly introduces the TRIO language, using the specification of the application analyzed in Section 5 as an example; Section 3 presents the inference rule on which our compositional framework for TRIO is based; Section 4 describes the PVS-based tool that supports the framework; Section 5 introduces some methodological considerations on the use of the compositional framework and shows how it can be applied to a simple, but meaningful, control system; Section 6 draws some conclusions and outlines future work in this line of research.

2 TRIO

TRIO [5,6] is a typed linear metric temporal logic enriched with object-oriented and modular features for writing specifications of complex systems.

[3] For the sake of space limit, we do not present extensively the literature related to this research. The interested reader can refer to [3] for a comparison with relevant related works.