Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Adaptability checking in complex systems

Emanuela Merelli [a,*], Nicola Paoletti [b], Luca Tesei [a]

[a] School of Science and Technology, Computer Science Division, University of Camerino, Via del Bastione 1, 62032, Camerino, Italy
[b] Department of Computer Science, University of Oxford, Parks Road, OX1 3QD, Oxford, UK

ARTICLE INFO

Article history:
Received 17 February 2013
Received in revised form 16 December 2014
Accepted 26 March 2015
Available online xxxx

Keywords:
Adaptive systems
State machine
Adaptability relations
Adaptability checking
S[B] model

ABSTRACT

A hierarchical approach for modelling the adaptability features of complex systems is introduced. It is based on a structural level S, describing the adaptation dynamics of the system, and a behavioural level B accounting for the description of the admissible dynamics of the system. Moreover, a unified system, called $S[B]$, is defined by coupling S and B. The adaptation semantics is such that the S level imposes structural constraints on the B level, which has to adapt whenever it no longer can satisfy them. In this context, we introduce weak and strong adaptability, i.e. the ability of a system to adapt for some evolution paths or for all possible evolutions, respectively. We provide a relational characterisation for these two notions and we show that adaptability checking, i.e. deciding if a system is weakly or strongly adaptable, can be reduced to a CTL model checking problem. We apply the model and the theoretical results to the case study of a motion controller of autonomous transport vehicles.

1. Introduction

Self-adaptive systems are particular systems able to modify their own behaviour according to their current configuration and the perception of the environment in which they operate. They develop new strategies in order to fulfil an objective, properly respond to changes of the environmental conditions or, more generally, maintain desired conditions.

From a broad viewpoint, self-adaptiveness is an intrinsic property of complex natural systems. Self-adaptation is a process driving both the evolution and the development of living organisms that adapt their features and change their phenotype in order to survive to the current habitat, to achieve higher levels of fitness and to appropriately react to external stimuli.

Nowadays, software systems are increasingly resembling complex systems; this motivates the development of methods for enabling software self-adaptiveness. Similarly to natural systems, *"Self-adaptive software evaluates its own behaviour and changes behaviour when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible"* [35]. Self-adaptive software finds application in fields like autonomic computing, service-oriented architectures, pervasive service ecosystems, mobile networks, multi-agent systems, and ultra-large-scale software systems [25], characterised by distributed, autonomous, interacting, heterogeneous, conflicting and evolvable subsystems.

In this work, we develop a formal hierarchical model for self-adaptive systems, where two fundamental levels are defined: the *behavioural level* B, which describes the admissible dynamics of the system; and the *structural level* S, accounting

[*] Corresponding author.
E-mail addresses: emanuela.merelli@unicam.it (E. Merelli), nicola.paoletti@cs.ox.ac.uk (N. Paoletti), luca.tesei@unicam.it (L. Tesei).

http://dx.doi.org/10.1016/j.scico.2015.03.004
0167-6423/© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

for the invariant features of the system that regulate the behaviour of the system. More precisely, both levels are modelled as state machines, but each state of the S level is associated with a set of *constraints*, i.e. logical formulas over observable variables of the B level.

An S state in the structural level represents a relatively persistent situation, a steady region for the B level, identified by the set of B states satisfying the constraints. Therefore S is a *higher order structure* whose states can be interpreted as sets of B states, and whose transitions can be viewed as mappings among sets of B states. The coupled model will be referred to as $S[B]$, in order to highlight the two basic levels that compose the system.

The $S[B]$ model is broadly inspired by a previous work of some of the authors, a spatial bio-inspired process algebra called *Shape Calculus* [3,4], where processes are characterised by a reactive behaviour B and by a shape S that imposes a set of geometrical constraints on their interactions and occupancy in the three-dimensional Euclidean space. Here, instead, the computational approach is shifted in a more general context, where S and B are coupled by a hierarchical relation defined on the structural constraints of the S level and the state space of the B level.

The adaptation semantics can be briefly described as follows. Consider an $S[B]$ system, let q be the current state of B and assume that it satisfies the constraints of the current S state. The adaptation is triggered whenever q cannot evolve because there is no next B state that satisfies the current S constraints. During adaptation, the $S[B]$ system attempts to evolve towards a new B state that satisfies a new S state, chosen among the successors of the current one. In this phase B is no more constrained by S. Adaptation terminates successfully when B ends up in a state that fulfils the new global situation represented by one of the admissible S states.

A first general introduction of the $S[B]$ model was given in [38] and [39] by the authors. In this work, we provide several novelties and improvements, most of them devoted to the *adaptability checking problem*, i.e. the automatic checking of the adaptation capabilities of a given $S[B]$ system. In particular, we define the notion of *weak adaptability*, possessed by an $S[B]$ system that is able to adapt along some of all its possible evolution paths. *Strong adaptability* requires that the $S[B]$ system is able to adapt along all its possible evolution paths. We formulate the notions of weak and strong adaptability as relations between the states of B and S and also in logical form, as Computation Tree Logic (CTL) formulas over the given semantics of an $S[B]$ system. Then, we formally prove the equivalence between the relational and the logical formulation of strong and weak adaptability (Theorems 1 and 2), showing that the adaptability checking problem can be reduced to a classical model checking problem. We also discuss the computational complexity of our approach. The effectiveness of the $S[B]$ approach for self-adaptive systems is demonstrated using a case study in the context of adaptive software systems, a model for a motion controller of autonomous transport vehicles in a smart airport. The same case study is used for exemplifying the notions of weak and strong adaptability and the adaptability checking problem.

The paper is organised as follows. Section 2 introduces the formalism and the syntax of the $S[B]$ model. Section 3 illustrates the application of the model to the example of adaptive motion controller. In Section 4 we give the operational semantics of an $S[B]$ system by means of a flattened transition system. In Section 5 we formalise the relations of weak and strong adaptation, which we equivalently characterise as CTL formulas in Section 6. Related works including adaptation features of $S[B]$, and conclusions are given in Section 7. Finally, proofs are presented in Appendix A.

2. A formal hierarchical model for adaptive systems

In the $S[B]$ approach, a model encapsulates both the behavioural (B) and the structural (S) level of an adaptive system. The behavioural level is classically described as a finite state machine of the form $B = (Q, q_0, \rightarrow_B)$ where Q is a set of B states, q_0 initial B state and $\rightarrow_B$ transition relation. The structural level is modelled as a finite state machine $S = (R, r_0, \mathcal{O}, \rightarrow_S, L)$ where R is a set of S states, r_0 is the initial S state, $\mathcal{O}$ is an observation function, $\rightarrow_S$ is a transition relation and L is a state labelling function. The function L labels each S state with a formula representing a set of constraints over an *observation* of the B states. Therefore an S state r can be directly mapped to the set of B states satisfying $L(r)$. Through this mapping, S can be viewed as a *second-order* structure $S = (R \subseteq 2^Q, r_0, \mathcal{O}, \rightarrow_S \subseteq R \times R)$ where each S state r is identified with its corresponding set of B states. An $S[B]$ system is the result of coupling the two levels S and B through the observation function $\mathcal{O}$ and an evaluation function $[[\cdot]]$, that maps each S state to the set of B states meeting the constraints.

The $S[B]$ adaptation is achieved by switching from an S state to another S state where a different set of constraints holds. During adaptation the B machine is no more regulated by the structural level, except for an adaptation invariant, called *adaptation invariant*, that must be fulfilled by the system while adapting. The system adapts by following any trajectory that is present at the B level that does not violate the invariant condition, which can be used as a safety condition if some activities of the B level must be avoided during adaptation.

In order to realise our notion of $S[B]$ adaptiveness, there must be some information flowing both from B to S and vice versa. In particular, the information from B to S is modelled here as a set of variables $A = \{a_1, \ldots, a_n\}$ called *observables* of the S on the B level. The values of these variables must always be *derivable* from the information contained in the B states. This makes our approach black-box-oriented, that is to say, S has not the full knowledge of B, but only some derived information.

Please cite this article in press as: E. Merelli et al., Adaptability checking in complex systems, Sci. Comput. Program. (2015), http://dx.doi.org/10.1016/j.scico.2015.03.004