Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 1164–1169

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Adaptive Technology
(WAT 2017)

Towards performance-focused implementations of adaptive devices

Paulo Roberto Massa Cereda[a,*], João José Neto[a]

[a]*Escola Politécnica, Departamento de Engenharia de Computação e Sistemas Digitais, Universidade de São Paulo*
Av. Prof. Luciano Gualberto, s/n, Travessa 3, 158, CEP: 05508-900 – São Paulo, SP – Brasil

Abstract

This paper presents instrumentation metrics for adaptive rule-driven devices as a means to obtain performance-focused implementations, from the underlying non-adaptive rule-driven device to the adaptive mechanism, as well as discussions regarding the adaptive behaviour and its corresponding operations, from theoretical and practical points of view.

Keywords: adaptive devices, instrumentation, implementation

1. Introduction

Adaptivity is the term used to denote a phenomenon in which a device spontaneously changes its internal behaviour in order to accommodate planned yet unexpected situations; these changes are triggered based solely on the device's own rule set and input stimuli, without any external interference[1,2]. A device is called adaptive if such feature is available to the model as a whole.

Although offering no computational power boost, adaptivity provides mechanisms for expressing abstractions more conveniently[1,3,4]. As a direct consequence, several model improvements are made possible and practically viable, such as complexity reduction[5], problem partitioning[6] and hierarchical solving[7], available at almost no sensible cost to the user[3].

Implementations of adaptive devices greatly vary, as well as the models themselves[3,8,9]. An early work of Cereda and José Neto[3] discussed potential bottlenecks and shortcomings on using common software engineering techniques as a means to implementing programs with adaptive characteristics, with drastic – and sometimes fatal – impacts on performance and stability. In this paper, we aim at extending the discussion to the adaptive mechanism, through fine-grained instrumentation of the adaptive behaviour and its corresponding operations, from theoretical and practical points of view.

Instrumentation is the capability of monitoring and recording a device behaviour and measuring performance during its life cycle[10]. It plays a crucial role in evaluation and testing procedures, as the collected data provide basis for achieving better performance and model improvements[11,12,13]. It is generally advisable to combine different metrics

* Corresponding author, +55 11 3091-5402.
E-mail addresses: paulo.cereda@usp.br (Paulo Roberto Massa Cereda)., jjneto@usp.br (João José Neto).

10.1016/j.procs.2017.05.390

in order to obtain a more comprehensive representation of such collected data, in an attempt to reduce bias [14,10]. However, producing traces incurs runtime overhead and therefore may interfere with the device's timing and perturb its behaviour [15], so instrumentation has to be kept to a minimum [16,13,17]. As to trace adaptive devices, we exposed their inner workings to analysis and gathered relevant data on queries and operations.

This paper is organized as follows: Section 2 formally introduces the concept of an adaptive rule-driven device. Section 3 presents presents the instrumentation metrics, as well as implementation aspects. Experiments and discussions are presented in Section 4. Finally, conclusions are presented in Section 5.

2. Adaptive rule-driven devices

This section formally introduces the concept of a general adaptive rule-driven device. It is important to observe that any non-adaptive rule-driven device may be enhanced in order to accommodate an adaptive behaviour while preserving its integrity and original properties [2]. The adaptive mechanism acts as simple extension to the underlying non-adaptive device.

Definition 1 (rule-driven device). A rule-driven device is defined as $ND = (C, NR, S, c_0, A, NA)$, such that ND is a rule-driven device, C is the set of all possible configurations, $c_0 \in C$ is the initial configuration, S is the set of all possible input stimuli, $\epsilon \in S$, $A \subseteq C$ is the subset of all accepting configurations (respectively, $F = C - A$ is the subset of all rejecting configurations), NA is the set of all possible output stimuli of ND as a side effect of rule applications, $\epsilon \in NA$, and NR is the set of rules defining ND as a relation $NR \subseteq C \times S \times C \times NA$. □

Definition 2 (rule). A rule $r \in NR$ is defined as $r = (c_i, s, c_j, z)$, $c_i, c_j \in C$, $s \in S$ and $z \in NA$, indicating that, as response to a stimulus s, r changes the current configuration c_i to c_j, processes s and generates z as output [2]. A rule $r = (c_i, s, c_j, z)$ is said to be compatible with the current configuration c if and only if $c_i = c$ and s is either empty or equals the current input stimulus; in this case, the application of a rule r moves the device to a configuration c_j, denoted by $c_i \Rightarrow^s c_j$, and adds z to the output stream. □

Definition 3 (acceptance of an input stimuli stream by a rule-driven device). An input stimuli stream $w = w_1 w_2 \ldots w_n$, $w_k \in S - \{\epsilon\}$, $k = 1, \ldots, n$, $n \geq 0$, is accepted by a device ND when $c_0 \Rightarrow^{w_1} c_1 \Rightarrow^{w_2} \ldots \Rightarrow^{w_n} c$ (in short, $c_0 \Rightarrow^w c$), and $c \in A$. Respectively, w is rejected by ND when $c \in F$. The language described by a rule-driven device ND is represented by $L(ND) = \{w \in S^* \mid c_0 \Rightarrow^w c, c \in A\}$. □

Definition 4 (adaptive rule-driven device). A rule-driven device $AD = (ND_0, AM)$, such that ND_0 is a device and AM is an adaptive mechanism, is said to be adaptive when, for all operation steps $k \geq 0$ (k is the value of an internal counter T starting in zero and incremented by one each time a non-null adaptive action is executed), AD follows the behaviour of an underlying device ND_k until the start of an operation step $k+1$ triggered by a non-null adaptive action, modifying the current rule set; in short, the execution of a non-null adaptive action in an operation step $k \geq 0$ makes the adaptive device AD evolve from an underlying device ND_k to ND_{k+1}. □

Definition 5 (operation of an adaptive device). An adaptive device AD starts its operation in configuration c_0, with the initial format defined as $AD_0 = (C_0, AR_0, S, c_0, A, NA, BA, AA)$. In step k, an input stimulus move AD to the next configuration and starts the operation step $k + 1$ if and only if a non-adaptive rule is executed; thus, being the device AD in step k, with $AD_k = (C_k, AR_k, S, c_k, A, NA, BA, AA)$, the execution of a non-null adaptive action leads to $AD_{k+1} = (C_{k+1}, AR_{k+1}, S, c_{k+1}, A, NA, BA, AA)$, in which $AD = (ND_0, AM)$ is an adaptive device with a starting underlying device ND_0 and an adaptive mechanism AM, ND_k is an underlying device of AD in an operation step k, NR_k is the set of non-adaptive rules of ND_k, C_k is the set of all possible configurations for ND in an operation step k, $c_k \in C_k$ is the starting configuration in an operation step k, S is the set of all possible input stimuli of AD, $A \subseteq C$ is the subset of accepting configurations (respectively, $F = C - A$ is the subset of rejecting configurations), BA and AA are sets of adaptive actions (both containing the null action, $\epsilon \in BA \cap AA$), NA, with $\epsilon \in NA$, is the set of all output stimuli of AD as side effect of rule applications, AR_k is the set of adaptive rules defined as a relation $AR_k \subseteq BA \times C \times S \times C \times NA \times AA$, with AR_0 defining the starting behaviour of AD, AR is the set of all possible adaptive rules for AD, NR is the set of all possible underlying non-adaptive rules of AD, and AM is an adaptive mechanism, $AM \subseteq BA \times NR \times AA$, to be applied in an operation step k for each rule in $NR_k \subseteq NR$. □