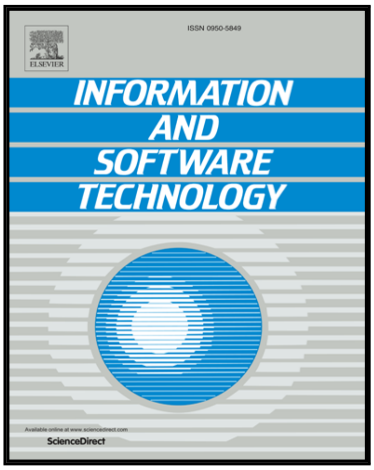


# Automatic Generation of Predictive Monitors from Scenario-based specifications

Pengcheng Zhang, Patrizio Pelliccione, Hareton Leung, Xuandong Li

PII: S0950-5849(17)30150-7
DOI: 10.1016/j.infsof.2018.01.014
Reference: INFSOF 5952

To appear in: *Information and Software Technology*

Received date: 22 February 2017
Revised date: 26 December 2017
Accepted date: 30 January 2018

Please cite this article as: Pengcheng Zhang, Patrizio Pelliccione, Hareton Leung, Xuandong Li, Automatic Generation of Predictive Monitors from Scenario-based specifications, *Information and Software Technology* (2018), doi: 10.1016/j.infsof.2018.01.014

# Automatic Generation of Predictive Monitors from Scenario-based specifications

Pengcheng Zhang[a,*], Patrizio Pelliccione[b], Hareton Leung[c], Xuandong Li[d]

[a] *College of Computer and Information, Hohai University, Nanjing* 211110*, P.R.China*
[b] *Chalmers University of Technology | University of Gothenburg, Department of Computer Science and Engineering, Sweden*
[c] *Department of Computing, Hong Kong Polytechnic University, HongKong, China*
[d] *State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing* 210046*, P.R.China*

## Abstract

**Context** Unpredictability and uncertainty about future evolutions of both the system and its environment may easily compromise the behavior of the system. The subsequent software failures can have serious consequences. When dealing with open environments, run-time monitoring is one of the most promising techniques to detect software failures. Several monitoring approaches have been proposed in the last years; however, they suffer from two main limitations. First, they provide limited information to be exploited at run-time for early detecting and managing situations that most probably will lead to failures. Second, they mainly rely on logic-based specifications, whose intrinsic complexity may hamper the use of these monitoring approaches in industrial contexts.
**Objective** In order to address these two limitations, this paper proposes a novel approach, called PREDIMO (PREDIctive MOnitoring). The approach starts from scenario-based specifications, automatically generates predictive monitors called MAs (Multi-valued Automata), which take into account the actual status and also the possible evolution of both system and environment in the near future, and enables the definition of precise strategies to prevent failures. More specifically, the generated monitors evaluate the specified properties and return one of the seven different values representing the degree of controllability of the system and the distance of the potential incoming failure. The translation from scenario-based specifications to MAs preserves the semantics of the starting specification.
**Method** We use the design and creation research methodology to design an innovative approach that fills highlighted gaps of state-of-the-art approaches. The validation of the approach is performed through a large experimentation with OSGi (Open Service Gateway Initiative) applications.
**Results** We present a novel language to specify the properties to be monitored. Then, we present a novel approach to automatically generate predictive monitors from the specified properties.
**Conclusions** The overall approach is tool supported and a large experimentation demonstrates its feasibility and usability.

*Keywords:* Scenario-based specifications, Property Sequence Charts, Run-time Monitor, Predictive Monitor

## 1. Introduction

Environments in which software-intensive systems are required to operate are increasingly evolving from static, closed and controllable to dynamic, open and uncontrollable due to the advent of new software paradigms such as Service-Oriented Architecture [1], Cyber-Physical Systems [2], Internetware [3], and the Internet of Things [4], just to name a few. The behavior of such software systems is jointly determined by their internal structure, and by the inputs received from both environment and end-users [5]. In this context, change is becoming the norm rather than the exception and the consequent unpredictability and uncertainty about future evolutions of both systems and environments are posing new challenges to software engineers [6, 7]. One of the most challenging problems is how to detect software failures in open environments as soon as possible. In general, software failure [8] means the undesirable or unacceptable external behavior of software when it is running. The continuous changes of environment and the system itself may lead to software behaviors that violate the original requirement specifications, resulting in software failures.

Traditional design-time verification techniques, such as testing and model checking, can ensure that software systems satisfy desired properties in closed or controllable environments. However, it is becoming difficult to cope with the continuous evolving environments and the execution states of systems at run-time in open environments. Run-time verification based on monitors has become the basic means of detecting software failures in such environments. The main idea of run-time verification is to synthesize a monitor, which can check whether run-time behaviors satisfy or diverge from desired properties [9, 10].

To design an efficient and effective monitor, the following

*I am corresponding author
*Email addresses:* pchzhang@hhu.edu.cn (Pengcheng Zhang), patrizio.pelliccione@gu.se (Patrizio Pelliccione), hareton.leung@polyu.edu.hk (Hareton Leung), lxd@nju.edu.cn (Xuandong Li)
*URL:* author-one-homepage.com (Pengcheng Zhang)