ARTICLE IN PRESS
JID: JSS [m5G;February 12, 2016;15:46]
The Journal of Systems and Software 000 (2016) 1–16
Contents lists available at ScienceDirect
The Journal of Systems and Software
journal homepage: www.elsevier.com/locate/jss

The verification of program relationships in the context of software cybernetics

Huaxiao Liu[a,b], Yuzhou Liu[a,b], Lei Liu[a,b,*]

[a] College of Computer Science and Technology, Jilin University, Changchun, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China, Changchun 130012, China

ARTICLE INFO

Article history:
Received 6 January 2015
Revised 19 January 2016
Accepted 21 January 2016
Available online xxx

Keywords:
Software cybernetics
Software verification
Control system

ABSTRACT

Software cybernetics aims at improving the reliability of software by introducing the control theory into software engineering domain systematically. A key issue in software verification is to improve the reliability of software by inspecting whether the software can achieve its expected behaviors. In this paper, the thought of software cybernetics is applied in the process of verification to address this issue and a nested control system is established. The proposed method verifies functional requirements in a dynamic environment with constantly changing user requirements, in which the program serves as a controlled object, and the verification strategy determined by software behavioral model (SBM) serves as a controller. The main contribution of this paper includes: (1) SBM is established in software design phase, and a concern-based construction approach is proposed, which starts from obtaining the software expected functionality extracted from a requirement text; (2) Program abstract-relationship model (PARM) is constructed basing on a process of gradual abstract to be a controlled object; (3) Feedback in a form of intermediate code is generated in the process of verification. The proposed method is validated by our case study.

1. Introduction

Software reliability is one of the most important criteria to measure the quality of software. In a dynamic environment, the software must be aware of the changes to make software work as expected with constantly changing user requirements which directly increases the demand for software reliability. To solve this problem, software cybernetics (Cai, 1995) and control-theoretic approaches (Abdelzaher et al., 2003) have been proposed.

Software cybernetics introduces theories of control to software engineering for improving the reliability of software, which can be defined as the interplay of software engineering and control theory (Cai et al., 2003). In software cybernetics, basic principles of control theory are applied to control the development process as well as the execution of software systems (Chen et al., 2009), answering questions such as why software needs to be controlled and whether or not it is possible for software to be controlled. Meanwhile, the roles of feedback mechanisms in software processes have been interpreted and improved by theory of software cybernetics. Besides having been studied to solve the problems in software testing by many researches, software cybernetics is exploited in other phases of software development. As Kenett said that “The expansion we envisage in the role of Software Cybernetics is in addressing, in an integrated way, the structure of any software based regulatory systems” (Kenett, 2011). The original scope of software cybernetics is expanded and turned into a natural platform for the interdisciplinary work and research to address challenges of software systems.

Software verification, which can detect many kinds of generic program errors even before a program is tested, is a discipline of software engineering. Its goal is to assure that the software fully satisfies all the expected requirements. And its progress requires the integration of multidisciplinary theories. Inspired by the notion of software cybernetics, we propose an approach termed as verification of program relationships relying on software behavioral model (VPRB). Our method treats the core problem in software verification of whether the software can achieve its expected behaviors as a control problem, formalizes the description of software expected behaviors and program, establishes a nested control system to manage the process of verification. In this control system, the outer layer is used to inspect whether the program could achieve its expected behaviors; the inner layer completes the concrete inspection of the internal relations in program. The changes of environment and user requirements are reflected in the outer

* Corresponding author. Tel.: +86 0431 85159373.
E-mail address: liulei@jlu.edu.cn (L. Liu).

http://dx.doi.org/10.1016/j.jss.2016.01.031

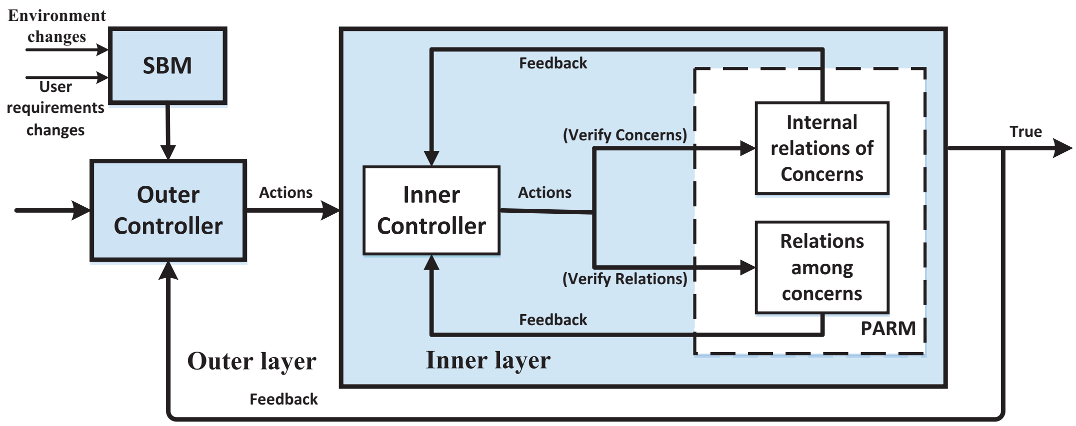

Fig. 1. Nested structure of VPRB based on software cybernetics.

layer controller and located in program by feedback, thus VPRB could cope better with a dynamic environment. In addition, the verification strategy/algorithm in the controllers can be changed to satisfy the need of different software verification in a proper perspective. Since our method is based on static analysis of program, the verification can also start right after implementing part of the program, rather than waiting until the entire program completes. This makes early detection of problems possible, making the work of correction convenient.

The paper is organized as follows. The overall approach of VPRB is presented in Section 2. Section 3 defines SBM and proposes the method to build it. Section 4 introduces the process of building PARM. The method of verification between the two models and the approach to generate feedback are introduced in Section 5. Section 6 shows an experiment. Its process and results validate the effectiveness of VPRB. Related work and discussion are presented in Section 7 and Section 8 respectively. Finally, conclusion is given in the last section.

2. The framework of nested feedback control system based on software cybernetics

Software cybernetics aims at achieving the control of software in each phase of development, such as testing and so on. Under the ideas of software cybernetics, we establish a control system to manage the process of software verification, treating that the program could satisfy the customer requirements, concretely expressed by the software expected behaviors, as the goal before control is applied.

There are three kinds of basic control in software cybernetics (Cai et al., 2003): Open-loop control, which means there is no feedback in control system, or the respond of the controlled object does not affect the subsequent actions; Closed-loop control (or Feedback control), the behavior of the controlled object is used in a feedback control strategy to determine subsequent control actions; Adaptive control, an advanced form of feedback control, the actions to be taken and the controller are affected by the responses of the controlled object. In order to make full use of the information generated in the verification process, we choose closed-loop control as the basic pattern to establish the framework of the control system in VPRB. On the whole, the process of VPRB is carried out in an iterative way. In each iteration, there includes the process of verification controlled by the closed-loop and the process of modification with practice guided by the results of verification. By using the approach of VPRB, we could improve the reliability of software system from the perspective of realizing the functionalities.

However, the architectures with only a single layer are not sufficient to address most of the control problems in software verification process. Therefore, a nested feedback control system is established to provide a better framework. In this framework, two important questions need to be considered.

Q1: How to describe software's expected behaviors and program in a way that could be applied in software cybernetics?

Q2: How to design feedback and how to use it in the process of software verification?

In this section, we will present the overall approach of the nested control system, and the detailed information of major components is provided in following sections. The nested structure of VPRB consists of outer layer and inner layer, shown as Fig. 1.

(1) Outer layer

The major tasks of outer layer include general process management of verification and requirements changes handling. The components are modeled and the input/output features among them are depicted by connecting arrows.

More specifically, firstly, SBM is the formalized description of software expected behaviors and established by requirements engineers in the requirements analysis phase. It includes the concerns and their relationships (details in Section 3), embodying the attributes that the software should contain for fully satisfying all the expected requirements. So that software expected behaviors could act on the control system and determine the verification strategy. This model is constructed by extracting the functional requirements of software.

The closed-loop is formed between the outer layer controller and the inner layer. The outer layer controller receives the feedback from the inner layer and gives subsequent control actions based on the verification strategy determined by SBM and the content of feedback. When errors are detected in the process, there may be two conditions: one is that the errors are produced in the process of writing program; the other is that the description of SBM is not in place because realistic questions have not been considered sufficiently.

Moreover, when requirements are changed by environment or user demands, SBM would be also changed correspondingly. With verification going, the control system will feed back the signals, which contain the contents needed to be modified and their locations in the program. It helps the software system to cope better with the complex environment.

(2) Inner layer

Inner layer focuses on controlling the concrete implementation of verification process, and it receives information from outer layer