Knowledge-Based Systems 000 (2017) 1–16

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A new framework for the verification of service trust behaviors

Jumana El-Qurna, Hamdi Yahyaoui*, Mohamed Almulla

Computer Science Department, Kuwait University, Kuwait

ARTICLE INFO

Article history:
Received 17 May 2016
Revised 19 December 2016
Accepted 6 January 2017
Available online xxx

Keywords:
Service
Model checking
Trust behaviors
Pushdown automata
Fixed-point Logic with Chop

ABSTRACT

We propose in this paper a model checking framework for service trust behaviors. We devise a new trust behavior model, which is a deterministic PushDown Automaton (*PDA*) based trust behavior model. This model is built based on the observations' sequences, which are derived from the interactions with services. Furthermore, we express the regular and non-regular trust behavior properties using Fixed point Logic with Chop (*FLC*). The model checking of service trust behaviors with respect to trust properties is performed using a symbolic *FLC* model checking algorithm. Finally, we present some experiments to assess the efficiency of the proposed algorithm.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

With the growing interest in using services, the verification of their functional and non-functional properties got special attention at the academic and industrial levels. Functional properties are related to the business logic of a service while non-functional properties are restricted to other requirements that are related to the service such as: availability, trust, security, reliability, scalability, etc. Many research initiatives tried to model and verify either separately the functional properties [7,25,51], and non-functional properties [10,60,66] or both kinds of properties [1,2,13].

Trust is one of the most important non-functional properties, which allows service providers and users to monitor and control their own service or other services' behaviors. A service trust behavior represents the conduct of a service during interactions with other services or users. Existing trust models generally focus on the calculation of trust values or the assessment of trust levels without taking into consideration long term behaviors of services. We pinpointed these issues in our previous works [63,64]. Long term trust assessment would lead without any doubt to a more accurate trust assessment.

One way to achieve such objective is to model service trust behaviors and perform model checking on such behaviors. Service trust behavior model checking is a valuable tool to analyze and assess the trust levels of services. Unfortunately, the modeling and verification of service trust behaviors is still in its genesis phase. In fact, there is no general framework that can be leveraged to perform a model checking of service trust behaviors. Such a framework requires the modeling of service trust behaviors, the description of trust properties and the proposal of an adequate model checking algorithm, which verifies if a service trust behavior satisfies or not a certain trust property.

We propose in this paper a model checking framework for verifying service trust behaviors. To the best of our knowledge, this is the first research initiative to do the model checking of trust behaviors with respect to trust properties. The service trust behavior model is represented as a finite transition system that explores all the possible trust behaviors of the service. The properties are classified into regular and non-regular properties, which denote typical trust behaviors. They are described using Fixed point Logic with Chop (FLC) logic. Our contributions can be summarized as follows:

- We introduce nine trust behaviors describing all the possible service trust behaviors, which are denoted by trust properties.
- We devise a trust behavior algorithm to model the service trust behaviors using pushdown automaton.
- We propose a model checking framework to verify whether the trust behavior model accepts the trust properties or not.

The rest of the paper is structured as follows. Section 2 is dedicated to the presentation of our service trust behaviors model checking framework. In Section 3, we discuss the related work. Section 4 is dedicated to the presentation of a formal specification of typical trust behaviors. In Section 5, we devise a trust behavior model for capturing a set of trust observations' sequences. Then, we define some FLC based trust properties in Section 6. In Section 7, we verify whether the trust behavior model satisfies

* Corresponding author.
E-mail addresses: jumana@cs.ku.edu.kw (J. El-Qurna), hamdi@cs.ku.edu.kw (H. Yahyaoui), almulla@cs.ku.edu.kw (M. Almulla).

http://dx.doi.org/10.1016/j.knosys.2017.01.011
0950-7051/© 2017 Elsevier B.V. All rights reserved.

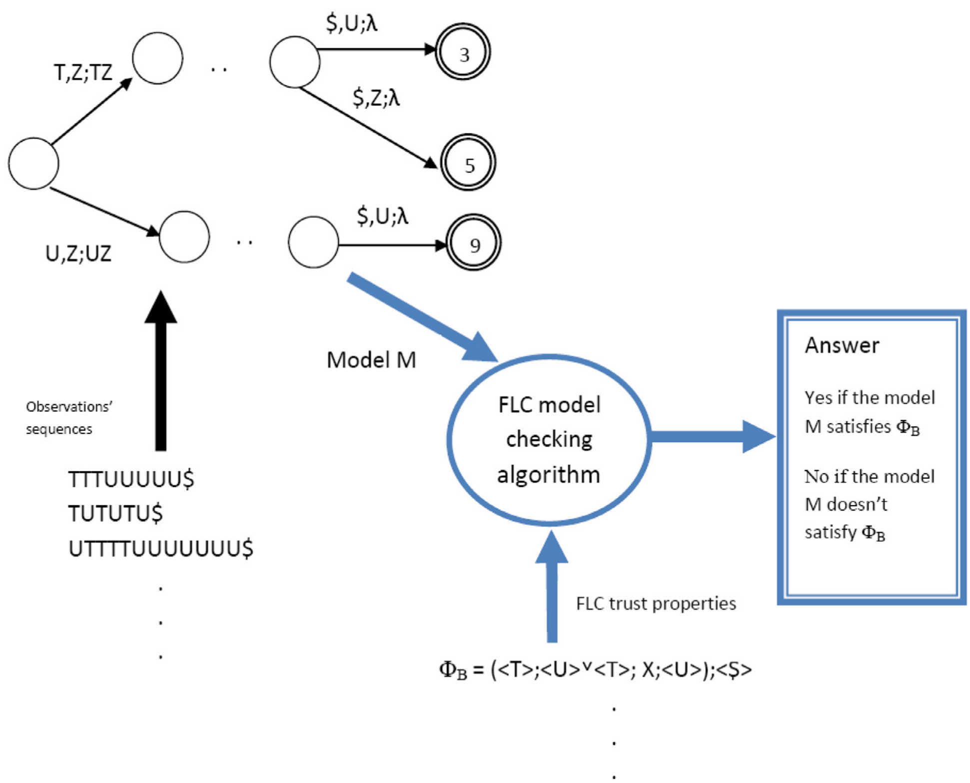

Fig. 1. The model checking framework for verifying service trust behaviors.

some trust properties using FLC model checking algorithm or not. Finally, in Section 8 we provide experiments that show how we model observations' sequences using our trust behavior algorithm and use them as input to mcflc tool to verify some trust properties. We discuss also the efficiency of the model checking algorithm.

2. Framework

The proposed framework is outlined in Fig. 1. It includes the following components:

- The model *M*: Model checking typically depends on a discrete model that is represented by a graph structure. The graph structure is annotated with more specific information. It assigns to each state *s* the atomic propositions *AP* that are satisfied in *s*. The atomic propositions *AP* are boolean expressions over variables, constants and predicate symbols. We are interested in the *deterministic PDA* based trust behavior model that captures the trust behaviors through the observations' sequences related to certain interactions between the user and the service. Model checking algorithms based on pushdown automata for various temporal logics have already been investigated by research community [9,28,53,58,59]. All these research initiatives proposed regular model checking algorithms based on pushdown automata that can only check the regular properties while the non-regular model checking algorithms based on pushdown automata are computationally intractable and as far as we know no one tried to tackle it. To overcome this issue, we generate the configuration graph of a *deterministic PDA*, which can be accepted by the model checker. Thus, the model *M* is the configuration graph of the *deterministic PDA* based trust behavior model that was derived from a set of observations' sequences.
- The property φ: Before applying the trust behavior model checker, it is necessary to state the trust behavior properties φ_i that the model *M* should satisfy. We are interested in the regular and non-regular trust behavior properties. Regular trust behaviors are behaviors that satisfy regular properties like "A red light is immediately followed by a yellow light" or "From any state, it is possible to get to a restart state". Non-regular trust behaviors are behaviors that depend on counting and can be represented using context free languages like "On every path the number of *a*'s so far never exceeds the number of *b*'s" and "On every path the number of the right parentheses "(" is equal to the number of the left parentheses ")"". More precisely, we identified in a previous work six regular trust behaviors: (1) Trustworthy (*T*), (2) Untrustworthy (*U*), (3) Betraying (*B*), (4) Redemptive (*R*), (5) Oscillating to Betraying ($O-B$), and (6) Oscillating to Redemptive ($O-R$). We also identify three non-regular trust behaviors: (1) Non-Regular Oscillating (*NRO*), (2) Oscillating to Trustworthy ($O-T$), and (3) Oscillating to Untrustworthy ($O-U$). The choice of using regular logics or non-regular logics depends on the properties to be analyzed. The regular logics capture regular properties. One of the common regular logics is μ-calculus [36], which forms a generalization of most temporal logics (*LTL, CTL* or *CTL*∗). The non-regular logics capture non-regular properties. The most common non-regular logics are: The linear time temporal logic *CaRet* [5], Non-Regular Propositional Dynamic Logic (PDL) [29,41], Modal Iteration Calculus (MIC) [18], Fixed point Logic with Chop (FLC) [39,42,45,50] and Higher-Order Fixed point Logic (HFL) [44,62]. Due to its similarity to context-free languages and its simplicity, Fixed point Logic with Chop *FLC* is chosen for analyzing the trust behaviors. Thus, the property φ is an *FLC* formula, which represents one of the regular and non-regular trust behaviors that are mentioned above.
- The verification: In this paper, we use a basic symbolic Fixed point Logic with Chop (*FLC*) model checking algorithm [6,40,43,62] that is based on the semantics of *FLC*.

Download English Version:

https://daneshyari.com/en/article/4946261

Download Persian Version:

https://daneshyari.com/article/4946261

Daneshyari.com