International Journal of Software Innovation
Volume 9 • Issue 4 • October-December 2021

Formal Analysis of Database Trigger Systems Using Event-B

Anh Hong Le, Hanoi University of Mining and Geology, Hanoi, Vietnam
https://orcid.org/0000-0002-0483-3195

To Van Khanh, VNU University of Engineering and Technology, Hanoi, Vietnam

Truong Ninh Thuan, VNU University of Engineering and Technology, Hanoi, Vietnam

ABSTRACT

Most modern relational database systems use triggers to implement automatic tasks in response to specific events happening inside or outside a system. A database trigger is a human readable block code without any formal semantics. Frequently, people can check if a trigger is designed correctly after it is executed or by manual checking. In this article, the authors introduce a new method to model and verify database trigger systems using Event-B formal method at design phase. First, the authors make use of similar mechanism between triggers and Event-B events to propose a set of rules translating a database trigger system into Event-B constructs. Then, the authors show how to verify data constraint preservation properties and detect infinite loops of trigger execution with RODIN/Event-B. The authors also illustrate the proposed method with a case study. Finally, a tool named Trigger2B which partly supports the automatic modeling process is presented.

KEYWORDS

Event-B, Modeling, Trigger, Verification

1. INTRODUCTION

Traditional database management systems (DBMS) are passive as they execute commands when applications or users perform appropriate queries. The research community has rapidly realized the requirement for database systems to react to data changes. Most modern relational databases include these features as triggers (or active rules) that monitor and react to specific events happening inside and outside of a system. They also use triggers to implement automatic tasks when a predefined event occurs.

DBMS usually have two types of triggers: data manipulation language (DML) and system triggers. The former is fired whenever the DML statements such as *deleting*, *updating*, and *insert* statements are executed, the latter is performed in case that system events or data definition language (DDL) ones occur. A trigger has the form of an Event-Condition-Action (ECA) rule informally written as "if a set of *events* occur and a set of *conditions* hold, then perform *actions*". It is made of a block of code and has syntax, for example, an Oracle trigger is similar to a stored procedure containing blocks of PL/

DOI: 10.4018/IJSI.20211001.oa1

SQL code. Trigger codes are understanding semantic and do not have any formal semantic. Trigger execution may lead to an infinite loop when triggers call each other or it may violate database system constraints. We can only check these properties after executing triggers. In fact, it is valuable if we can show that trigger execution is correct at design phase because it reduces development cost for database design. Thus, a formal framework for modeling and verifying database triggers is desirable.

Many researchers have been working on analyzing triggers (or active rules). The research results of Lee and Ling (1198, 1999), proposed in the early 1990s, transformed ECA rules to some types of graphs and applied various static analysis techniques to check properties such as *redundancy*, *inconsistency*, *incompleteness*, and *termination*. Baralis (1999) proposed a technique, based on relational algebra, to check if active rules are terminated or confluent. Other results Choi et al (2006b); Chavarría-Báez and Li (2007) addressed both *termination* and *confluence* properties using model checking techniques. One important property that has not received much attention is data constraint property of a system. A terminated trigger still can cause critical problems if it violates data constraints. Furthermore, a method or an approach with supporting tools, which is feasible to apply in database development to check both data constraints and infinite loops, is also desirable.

Our previous work Le and Truong (2013) initially proposed to use Event-B to formalize and verify a database triggers system at design phase. The main idea of the method comes from the similar structure and working mechanism of Event-B events and database triggers. We presented a set of translation rules to translate a database system including triggers to an Event-B model. In this paper, we make the translation rules more precise when encoding the body of trigger. With the proposed modeling method, we can formally check if a system preserves the data constraints and find infinite loops by proving the proof obligations of the translated Event-B model. The advantage of our method is that a database system including triggers and constraints can be modeled naturally by Event-B constructs such as *invariants* and *events*. As far as we know, this paper reports the first concrete result of analyzing a database system with DML triggers using formal method. We also developed a tool called Trigger2B which partly supports automatic modeling process with the RODIN. We introduce an algorithm to construct an Event-B model from a syntax tree of triggers written in SQL. In the supporting tool RODIN, almost proofs are discharged automatically, hence it reduces complexity in comparison with manual proving. The tool makes the proposed method feasible in database development process.

The remainder of this paper is organized as follows. Section 2 provides basic knowledge of database triggers and Event-B. In Section 3, we present our proposed method to model and verify database systems including triggers. Section 4 introduces a scenario of a human resource management application to demonstrate the method in detail. We present the tool Trigger2B, which supports for partly automatic translation in Section 5 and we summarize related work in section 6. We conclude the paper and present the future work in Section 7.

2. PRELIMINARIES

In this section, we first briefly introduce database triggers and their SQL syntax. Then we give an overview of Event-B formal method.

2.1. Database Triggers

A relational database system, based on the relational model, consists of collections of objects and relations, operations for manipulation and data integrity for accuracy and consistency. Modern relational database systems include active rules as database triggers which response to events occurring inside or outside of database.

A database trigger is a block code that is automatically fired in response to a defined event in a database. The defined event is related to a specific data manipulation of the database such as inserting, deleting, or updating a row of a table. Triggers are commonly used in some cases: to audit the process,