Model Driven Engineering for Quality of Service Management: A Research Note on the Case of Real-Time Database Management Systems

Salwa M’barek, RIADI Laboratory, ENSI, Tunis, Tunisia
Leila Baccouche, INSAT, University of Carthage, Tunis, Tunisia
Henda Ben Ghezala, RIADI Laboratory, ENSI, Tunis, Tunisia

ABSTRACT

Real-time applications managing a large number of real-time data require the use of Real-time Database Management Systems (RTDBMS) to meet temporal constraints of both real-time data and transactions. However, a RTDBMS has a dynamic workload and may be frequently overloaded since the arrival times and workloads of user transactions are unpredictable. Therefore, Quality of Service management solutions have been proposed to guarantee the stability of RTDBMS even during unpredictable overload periods. While effective, the design and reuse of these solutions is challenging because they are not formally modeled and there is no tool neither a methodology that helps us design such solutions. To address these issues, the authors propose a design framework based on the Model-Driven Engineering approach providing a modeling architecture, a strategic methodology and a software tool to support modeling and reusing such solutions. The framework is implemented and tested for a real Qos management solution.

KEYWORDS
Metamodeling, Model-Driven Engineering, Model Transformations, Quality of Service, Real-Time Database Management Systems, Reuse

INTRODUCTION

In recent years, Real-Time Applications (RTA) providing real-time services are becoming more and more expanding such as Manufacturing Execution Systems (MES), web-based multimedia applications like online Video on Demand and streaming, Vehicular Cyber-Physical Systems (VCPS), meteorology applications. These applications manage big volumes of real-time data and have to meet both logical and temporal constraints on both real-time data and transactions. For this aim, they use Real-time Database Management Systems (RTDBMS) which are more suitable to this context because they combine real-time systems and database management systems features (Bestavros, Lin & Son, 2012).

Applications often express quality of service requirements that have to be integrated in the database system and handled through quality of service (QoS) Management Solutions, in order to guarantee the stability of RTDBMS even during unpredictable overload periods.

Many RTA need specific QoS constraints and parameters which cannot be addressed by existing solutions which are application-dependent. Given the complexity of the solution design and their effectiveness, their reuse is very useful and beneficial. However, it is challenging because QoS
Management Solutions are not formally modeled but simply described in a natural language. In addition, there is no methodology or tool helping designers in this context.

In our work, we provide models of these solutions that allow the automated processing and the model reuse in order to extend existing solutions or design a new one through the composition or transformation of generated models. The approach we have proposed is based on the Model-Driven Engineering (MDE) (Brambilla, Cabot & Wimmer, 2006) that best suited our requirements because it provides a generic framework for metamodeling and model transformations.

The paper is organized as follows. We begin by describing QoS management in RTDBMS and some related work. We then illustrate our modeling work for QoS management solutions after introducing the MDE approach. The framework consisting in a logical architecture, a design methodology and model transformations is detailed afterwards. Finally we present the implementation and experiments on a real QoS Solution. We end this paper with the conclusion and future works.

**QoS MANAGEMENT IN RTDBMS**

Real-time transactions manipulating real-time data can be update or user transactions. Update transactions have to periodically update real-time data which have validity periods beyond which they become not fresh. The consequences of accessing data outside their validity periods depend on particular requirements of the application and data semantics. User transactions may access real-time data and update non real-time data. They must be processed within their deadlines and use fresh data (Ramamritham, Son & Dipippo, 2004). Moreover, they have unpredictable arrival and execution times. In this context, the RTDBMS may face unpredictable overload periods over time, during which its performances may be downgraded and many transactions may miss their deadlines.

To address this problem, many QoS Management Solutions have been proposed and most of them have used the Feedback Control Scheduling Architecture FCSA (Lu, Stankovic, Tao & Son, 2002). The first sub-section explains how to specify the QoS in RTDBMS and the operating principle of the FCSA architecture. The second sub-section presents some QoS solutions for RTDBMS.

**Quality of Service Specification**

The QoS specification is given by the database administrator through a set of QoS parameters and their reference values to express the desired performance of the RTDBMS. In the literature, QoS parameters are classified into QoT parameters and QoD parameters. RTDBMS can ensure the quality of data (QoD) and/or the quality of transactions (QoT). The QoD relies on providing fresh data that are periodically updated in order to accurately reflect the state of the real world. The QoT implies transactions scheduling within deadlines in order to give reliable results (Ramamritham, Son & Dipippo).

For instance, in (Amirijoo, Hansson & Son, 2004), the deadline Miss Ratio (MR) calculates the percentage of transactions that have missed their deadlines according to all executed transactions. It is considered as a QoT parameter. The Maximum Data Error (MDE) considered as a QoD parameter, measures the deviation of the stored values in the database relative to the real world values making this solution ensure both QoD and QoT.

The QoS is specified in terms of steady and overload state. In case of overload, a bounded overshoot of reference values is tolerated. Several QoS parameters, can be used to express the desired performance.

The most common architecture used by these solutions is the Feedback Control Scheduling Architecture (Lu, Stankovic, Tao & Son). It monitors the QoS and dynamically adjusts it to guarantee...