The Database Version Approach: Overview and Potential Directions

In tribute to Geneviève Jomier (1948 - 2018)

Talel Abdessalem
LTCI, Télécom ParisTech, Université Paris-Saclay
Paris, France
talel.abdessalem@telecom-paristech.fr

Claudia Bauzer Medeiros
University of Campinas, Institute of Computing (IC - UNICAMP)
Campinas, Brazil
cmbm@ic.unicamp.br

Wojciech Cellary
Poznan University of Economics
Poznan, Poland
cellary@kti.ue.poznan.pl

Stéphane Gançarski
LIP6 - U.P.M.C
Paris, France
Stephane.Gancarski@lip6.fr

Khaled Jouini
MARS Research Lab LR17ES05, ISITCom, University of Sousse
Sousse, Tunisia
khaled.jouini@isitc.u-sousse.tn

Maude Manouvrier
Université Paris-Dauphine, PSL Research University, CNRS UMR [7243] LAMSADE
Paris, France
maude.manouvrier@dauphine.fr

Marta Rukoz
Université Paris-Dauphine, PSL Research University, CNRS UMR [7243] LAMSADE
Paris, France
marta.rukoz@dauphine.fr

Michel Zam
Université Paris-Dauphine, PSL Research University, CNRS UMR [7243] LAMSADE
Paris, France
zam@dauphine.fr

ABSTRACT

In 1990, W. Cellary and G. Jomier proposed the Database Version (DBV) approach, which allows to manage multiversion databases — those in which several versions of a set of data items coexist. Ever since, its model, theory and algorithms have been adopted in a multitude of research initiatives and publications, and been applied to a variety of applications, in particular those in which there is a need for keeping track of parallel or (spatio)-temporal evolution of states of the world. This article presents an overview of the DBV approach, and some of the associated research initiatives throughout three decades, pointing out new potential directions. It has been written in tribute to Geneviève Jomier, Prof. Emeritus of The Université of Paris-Dauphine, who left us in March 2018.

1 INTRODUCTION

This article presents the *Database Versions* (DBV) approach, proposed by W. Cellary and G. Jomier in 1990 in [10].

Formally, a database is monoversion, representing only one state of the modeled world; each data item has a single value. In the DBV approach, a multiversion database brings together several states of the world, these states being variants or evolutions in time of the modeled world. Each data item has, in this case, several versions, the value of that item potentially changing from one version to another. Each Database Version represents a consistent state or a configuration of the modeled world [17]. DBVs can be applied to any type of data, whether images [35, 36], documents [5] or spatio-temporal data [42, 43, 48].

As will be seen throughout this text, the DBV model can be adopted in a variety of situations, whenever there is a need for managing many states of the world. Though this may seem to be an obvious statement, since it models versions (and thus states of the world), it is simple and generic. Through the DBV model, versions can be managed through a small set of compact data structures, simplifying maintenance of co-existing states, and speeding up rollback to any database state. For this reason, it was implemented in many computing platforms, for several purposes, in a variety of contexts.

The DBV model has originated several research initiatives, including the ones presented in a variety of papers - e.g., [2, 5, 17, 24, 32, 33, 36–39, 48, 49, 52]. Several PhD thesis in computer science [1, 18, 22, 35, 42, 50], supervised by G. Jomier, are based on this model. The model has also inspired thesis [41] and research projects - e.g., [13, 31, 44], in other institutions.

Section 2 presents the DBV approach and Section 3 discusses a few of its applications. Section 4 concludes the paper, presenting new potential research directions which can still be exploited.

2 DBV APPROACH

This section gives an overview of the DBV approach and of its subsequent generalizations.

2.1 Overview of the concepts

A DBV is identified by v. A multiversion data is associated with an immutable identifier, d. A DBV contains exactly one *logical version* d_j of each multiversion data d, having an identifier and a value val.