Information and Software Technology 92 (2017) 30–48

Contents lists available at ScienceDirect

Information and Software Technology

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

Developing software systems to Big Data platform based on MapReduce model: An approach based on Model Driven Engineering

S. Sousa Osvaldo Jr., Denivaldo Lopes*, Aristófanes C. Silva, Zair Abdelouahab

Federal University of Maranhão - UFMA, São Luiz, Brazil

ARTICLE INFO

Article history:
Received 10 November 2016
Revised 12 June 2017
Accepted 6 July 2017
Available online 12 July 2017

Keywords:
Model Driven Engineering
Big Data
Framework
Metamodels

ABSTRACT

Context: The need to analyze a large volume and variety of data for the purpose of extracting information has been promoting investments in Big Data, e.g., for storage, analysis and, more recently, methodologies and approaches for software system development for Big Data platforms. The application of software engineering for Big Data is recent and emerging, so in the literature we find a number of challenges and opportunities related to Big Data, but few practical approaches.
Objective: In this paper, we propose a practical approach based on MDE (Model Driven Engineering) to support the semi-automated development of software systems for Big Data platform that use MapReduce model.
Method: The proposed approach consists of framework, process, metamodels, visual Alf, transformation definitions written in ATL and Eclipse IDE plug-in. The proposed framework uses concepts of MDE, Weaving and software development based on Y. Our proposed process guides the use of our approach. A graphical notation and extended metamodel for Alf (i.e. visual Alf) assign executable behavior for UML or DSLs. An Eclipse IDE plug-in implements our approach.
Results: We show the applicability of the proposed approach through an illustrative example.
Conclusion: Our approach brings a contribution because the development of software systems is assisted by models which preserves the business logic and adds Big Data features throughout the development process.

1. Introduction

In the last decades, technological innovations in hardware and software, together with the expansion of Internet access, enabled the world to become increasingly connected. In November 2015, more than 3 billion users were connected to the Internet in the world, approximately 47% of the world population[1]. In May 2013, this scenario was just over 2 billion users, approximately 38% of the world population [2]. In two years, there was a growth of 23% of users connected.

This volume of Internet access quickly generated a volume and variety of data that large computers alone do not have the capacity to store or analyze. The growing search for relevant information made large companies realize that this huge volume of data could bring a positive difference in the current global market. In this context, the term Big Data is used to conceptualize this large volume of data that has different characteristics and accelerated growth [3,4].

Sener et al. [5] demonstrate that massive interest in Big Data is recent (mid-2012), and that interest is taking exponential growth. In recent years, massive investment from scientific community to provide solutions to the most diverse aspects of Big Data is notorious, for example: visualization [6–8], data analysis [9–11], storage [12–14], and security [15–18]. Fang et al. [19], confirm interest in Big Data by presenting a discussion of challenges and solutions in industry and academia from the perspective of engineers, computer scientists and statisticians, while Lee et al. [20] present challenges and opportunities related to geospatial data.

An emerging and recent concern related to research on Big Data platform is how to apply concepts, methods and techniques derived from Software Engineering on this platform. This concern began to be raised by the community with more emphasis in 2015. For example, Otero et al. [21] and Madhavji et al. [22] present the challenges related to requirements, analysis, design and implementation tasks without offering concrete solutions, but rather only suggestions on how to solve them. The work of Anderson et al. [23] and DeLine et al. [24] presents similar challenges, but they

* Corresponding author.
E-mail addresses: oss.junior@ufma.br (S.S. Osvaldo Jr.), denivaldo.lopes@ufma.br, denivaldo.lopes@gmail.com (D. Lopes), ari@dee.ufma.br (A.C. Silva), zair@dee.ufma.br (Z. Abdelouahab).

http://dx.doi.org/10.1016/j.infsof.2017.07.006
0950-5849/

point out the importance of offering solutions that aid in the development process, such as tools or frameworks. While, Bagheri et al. [25] raises questions about theoretical or practical work on appropriate methods, techniques, processes and tools for software engineering applied to Big Data.

One of the problems found in the context of Software engineering applied to Big Data platform is the lack of solutions (frameworks, tools, techniques) that fully support software development activity for this platform in an integral manner and with some level of automaticity [25]. Development activity is composed of tasks such as analysis, design and implementation, that is, tasks that range from problem modeling to full source code creation.[1]

Analysis and design tasks typically use models to understand, describe, and document the problem. Recently, Chen et al. [26] presented a concrete solution that supports problem modeling for Big Data, but did not offer implementation-level solutions. The implementation task, in turn, aims to produce the complete source code. This task for Big Data would be complex because there are many variables involved (network, resource sharing, parallelism, among others), but some solutions that allow encapsulation of this complexity were proposed by Apache in the form of frameworks such as Hadoop's MapReduce [27] and Spark [28]. These frameworks implement or are based on the MapReduce model.[2] This model is based on the divide-to-conquer paradigm and can be summarized in two phases: Map and Reduce, which will be explained in Section 2.3.

Isolated solutions for development tasks for Big Data MapReduce exist, but these solutions cause a rupture between what is modeled and the source code created. This rupture harms documentation, does not preserve investments,[3] generates rework and reduces productivity in maintenance and upgrades. In this context, it is necessary to think about solutions that automatically or semiautomatically support development tasks for MapReduce of Big Data.

In the software engineering area, the Model Driven Engineering (MDE) provides support, through the use of models, to improve productivity in software development, preserve the business logic and provide quality aspects such as maintenance and interoperability [29]. The emergence of MDE made the models no longer seen only as documentation, but like essential artifacts within the software development process [30,31]. The software creation based on models was possible by using some technologies, e.g. metamodel, model transformation and code generation [30]. MDE has been used for various purposes, e.g. security [32], software testing [33,34] and database merging [35].

We propose an approach based on MDE, which uses concepts of weaving and software development based on process in Y, to support the semi-automated development of software systems for Big Data platform that use MapReduce model. It is important to emphasize that we are not proposing an algorithm, but rather a solution that allows to assist in the development activities of software for Big Data based on MapReduce model using existing frameworks, models and transformations between models.

Some of the existing solutions for software development, for example modeling tools, try to be generalist, that is, they seek to be applied in all domains. This generality often makes the proposed solutions very limited. Thus, there is a lack of investments in more specific solutions. Because our proposed framework is specific to MapReduce model, it potentiates the development for Big Data platform using MapReduce model, making some tasks automatic, which would be otherwise performed manually. It also ensures investments preservation, corrective maintenance easiness and evolution of created software.

This paper is organized as follows. Section 2 provides an overview of MDE and Big Data. In Section 3, we present the proposed approach, including the framework, process, metamodels, visual Alf and transformation definitions. In Section 4, we present the prototype of our approach. Section 5 presents an illustrative example developed according to our proposed approach. In Section 6, we present some related work and a comparison of our approach and other approaches to support the software developement based on Big Data platform. Finally, the conclusions are presented in Section 7.

2. Overview

In this section we present an overview of MDE, Big Data and development technique for Big Data.

2.1. Model driven engineering

The software creation process is not a trivial task because it involves empirical factors and it is constantly evolving. The advances achieved over more than four decades in software development were only possible because professionals have been developing research in software engineering [30].

The software engineers responsible for the advances in software engineering have been researching new methods and tools to enable software development to be done with speed, quality and above all with low investment [30]. The best known methods according to Pressman [36] are Prescriptive Software Process (PSP) and those based on Agile Principles (AP). The former process prescribes a series of activities, tasks and artifacts to help developers build software within a life cycle, while the latter prescribes a minimum of steps, because they are based on the values of the agile manifest [37]. This manifest assumes that customer participation, strong team interaction, software running and rapid response to changes are more effective than documentation and excessive prescriptions.

The fact is that although there are advances in software development, through methods such as PSP and AP, there are still worrying factors within the software construction because some projects do not meet satisfactorily requirements or lack quality. Others are not delivered on time, exceed budgets and few can keep up the dynamism, in terms of changes, currently required. Some issues can impact development processes such as synchronism loss and, sometimes, lack of standardization in the produced artifacts within all the development life cycle, over several iterations.

MDE aims to support the management of the software development complexity and express efficiently domain concepts [38]. MDE can provide this support by raising the abstraction level using models, notations and transformation rules [39]. The pillars supporting the MDE approach are models, metamodels, modeling language and model transformations [30,39–41]. MDE can assist in managing the software development life cycle thanks to models, supporting the development, maintenance, evolution of a software system preserving the investments and assuring its quality.

2.2. Big Data

Cheikh et al. [4] made a survey about the definition given to Big Data by many authors and they conclude that Big Data definition

[1] Full source code refers to the skeleton of methods plus its behavior (logic) of the method.

[2] The framework used as a processing solution by the Hadoop platform takes the same name as the template MapReduce.

[3] Preserve investments in this work refers to business logic completely embedded in models. Usually, in modeling tasks only a macro view of business logic is represented through methods definition, since the models do not offer a standardized and homogeneous way of including a micro view of business logic that is represented by the method behavior definition.