The state of big data reference architectures: a systematic literature review

Abstract

This template helps you to create a properly formatted LaTeX manuscript.

Keywords: `elsarticle.cls`, LaTeX, Elsevier, template

2010 MSC: 00-01, 99-00 s

1. Introduction

The rapid development of software technologies, the proliferation of digital devices and networking infrastructure of today, have by and large, augmented user's capability to generate data [1]. In the age of information, users are unceasing generators of structured, semi-structured, and unstructured data that if collected and crunched correctly, may reveal game-changing patterns [2].

The unprecedented proliferation of data have emerged a new ecosystem of technologies; one of these ecosystems is big data (BD)[3]. BD is a term emerged to describe large amount of data that comes in various forms from different channels. Within the years, BD has attained a lot of attention from academia and industry, and many strive to benefit from this new material. Howbeit, adopting BD requires the absorption of great deal of complexity and many traditional systems cannot cope with characteristics of this domain.

A recent survey published by Databricks in partnership with MIT Technology Review Insights, stated that only 13% of companies excel at delivering on their data strategy [4]. In the same vein, Vintage Partners highlighted that only 24% of companies have successfully adopted BD [5]. Sigma computing report presented that 1 in 4 business experts have given up on getting insights they needed because the data processing took too long [6]. Moreover, Gartner approximated that only 20% of companies have successfully adopted BD.

Some of the most highlighted challenges of BD is 'lack of business context', 'organizational challenges', 'BD architecture', 'data engineering', 'rapid technology change', and 'lack of talent' [7]. Whereas similar issues may exist in other domains, it is exacerbated when it comes to BD systems. This is due the inherent complexity of BD engineering, the need for real-time processing, the scalability requirement of these systems, and the sensitivities around data.

Today, majority of BD systems are designed underlying ad-hoc and complicated architectural solutions [8], that do not seem to adhere to similar patterns. This will challenge software architects to design a suitable solution for any given context, creates a foundation for an immature architectural decision, and does not promote the growth and development of BD systems as a whole.

Therefore, since the approach of ad-hoc design to BD systems is undesirable and leaves many engineers in the dark, there is a need for more software engineering research for BD systems. To this end, this study presents a systematic literature review (SLR) on BD (BD) reference architectures (RAs).

2. Why reference architectures?

Conceptualization of the system as an RA, helps with understanding of the system's key components, behavior, composition and evolution of it, which in turn affect quality attributes such as maintainability, scalability and performance [9]. Therefore RAs can be a good standardization artefact and a communication medium that not only results in concrete architectures for BD systems, but also provide stakeholders with unified elements and symbols to discuss and progress BD projects.

This approach to system development is not new to practitioners of complex system. In software product line (SPL) development, RAs are utilized as generic artifacts that are instantiated and configured for a particular domain of systems [10]. In software engineering, IT giants like IBM have referred to RAs as the 'best of best practices' to address complex and unique system design challenges

[9]. In other international standardization, RAs have been repeatedly used to standardize an emerging domain, a good example of this is BS ISO/IEC 18384-1 RA for service oriented architectures [11].

3. State of the art

Despite the undeniable benefits of RAs, and their potential to solve some of the complex issues of BD systems, we think that this area is underdeveloped and needs more attention from both academia and practice. This insight is derived from our preliminary systematic review in academia, and a search for available big data RAs ([2]).

To the best of our knowledge, one of the most comprehensive BD RA published, is the National Institute of Standards and Technology (NIST) BD RA. This RA is published by Big Data Public Working Group (NBD-PWG) with large set of contributors from academia, industry, non-profit organizations, agents, and government representatives. This was announced as an initiative from White house in March 2012, and the the RA was published under the title 'NIST Big Data Interoperability Framework: Volume 6, Reference Architecture' in October 2019.

Given the substantial investment on BD RAs, one might infer the value of these artifacts, and this can in turn highlights the necessity for more research in this domain. Another factor that worths mentioning is how vaguely the phrase 'reference architecture' is defined and institutionalized. For instance, the difference between a 'concrete architecture' and an RA is hardly discussed, and different domains seem to have defined the artifact slightly differently. For instance, Cloutier et al ([9]) defined RAs as 'Reference Architectures capture the essence of existing architectures, and the vision of future needs and evolution to provide guidance to assist in developing new system architectures'. This definition is derived from the system engineering domain and by the means of collaborative forum from Steven's institute of technology.

In another effort, Muller et al ([12]) defines RA as 'artifacts that captures the essence of architecture of a collection of systems. This definition is driven from the product line engineering domain'. Moreover, the difference between RAs and concrete architectures is rarely discussed. Another definition by Bass et al ([13]) stated that 'A reference architecture is a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them'.

Angelov et al ([14]) defined RAs proposed that 'A reference architecture is a generic architecture for a class of information systems that is used as a foundation for the design of concrete architectures from this class'. Although different authors may have defined RAs with different syntax, the essence remains the same: to reuse the software engineering knowledge for a class of systems, particularly in relation to architecture.

Given the failure rate of BD projects, we posit RAs as potential solution to facilitate system development and BD architecture, and aim to explore this area through a systematic literature review. Up to date, there's only one SLR that explored this area ([2]), which is outdated, suffers from methodological clarity, and is published as a conference paper, which implies lack of detail.

Based on this, the objective of this review is to find and collate the BD RAs available from the body of evidence, highlight their architectural commonality and point out the limitations. This study can be considered a useful primer for practitioners or academics who are interested in partaking in a BD project.

The research questions are formulated as the following;

1. What are current BD RAs available in academia and industry?
2. What are major architectural components of these BD RAs?
3. What are the limitations of current BD RAs?

4. Review Methodology:

This research follows the guidelines of PRISMA ([15]). In addition, we adopted PRISMA-S ([16]) to improve our search strategy and lastly we have used Barbara et al's guidelines for evidence based software engineering and systematic reviews [17]. Although PRISMA is a comprehensive guidelines on conducting a systematic literature review, it is derived from the healthcare community and sometimes makes assumptions that may not be relevant to software engineering and information system researchers. Barbara et al [17] have translated many of these assumptions to the domain of software engineering and have included many guidelines for lone researchers and projects with small number of researchers.

We have therefore utilized PRISMA as the underpinning of our research design, with complementary studies to reduce bias, improve transparency and systematiticity. SLR has been chosen because it is a qualitative research methodology that is aimed at driving knowledge and understanding about the subject matter and the elements around it. Besides, SLR provides a transparent and reproducible procedure that elicits patterns, relationships, trends, and delineates the overall picture of the subject [18].

The main objective of this study is to assess the current state of BD RAs, identify their major architectural components, point out fundamental concepts and discuss their limitations. This objective is achieved in four phases. In first phase, research questions are stated, literature are identified and pooled, and exclusion and inclusion criteria are defined. In second phase, literatures are assessed for their quality based on inclusion/exclusion criteria and relevance to research questions. Thirdly, selected pool of literature is coded based on research questions. Lastly, findings are synthesized by the means of thematic synthesis, and themes realized are depicted.

5. Improvements

1. The current writing style looks like a summary description, lacks new insight on the topic. The overall contribution needs to be enhanced.
2. The inclusion criteria and exclusion criteria are ambiguous and questionable. Is it possible for readers to reproduce this study according to these criteria? Did the author perform the reliability and validity tests? The

author needs to provide more detail about the review methodology.

3. In general, each larger-scale system requires a more understanding of architectural components, owing largely to the complex nature of system architects. However, I cannot find a case that the authors demonstrate the uniqueness of BD systems, and the actual development challenges in BD systems.
4. The findings yielded by investigating the research questions of this SLR should constitute many discussion points around the research and practice of BD systems. However, the manuscript is completely missing a discussion section. One should expect that the results of SLR can inform the current knowledge and provide several research directions for future research.
5. Last, one of the core challenges with the paper is to situate it within an ongoing scholarly conversation. The authors currently reference a fairly diverse set of papers, but remain at a fairly abstract level when it comes to elaborating how your work builds upon and expands existing work. In turn, this makes it difficult to appreciate theoretical implications of your work.

References

[1] B. Bashari Rad, N. Akbarzadeh, P. Ataei, Y. Khakbiz, Security and privacy challenges in big data era, International Journal of Control Theory and Applications 9 (43) (2016) 437–448.

[2] P. Ataei, A. T. Litchfield, Big data reference architectures, a systematic literature review (2020).

[3] B. B. Rada, P. Ataeib, Y. Khakbizc, N. Akbarzadehd, The hype of emerging technologies: Big data as a service (2017).

[4] Databricks.
URL https://databricks.com/

[5] N. Partners, Big data and ai executive survey 2021 (2021). URL https://www.supplychain247.com/paper/bi_data_and_ai_executive_survey_2021/pragmadik

[6] S. Computing, Bridging the gap between data and business teams (2020). URL https://www.sigmacomputing.com/resources/data-language-barrier/

[7] B. B. Rad, P. Ataei, The big data ecosystem and its environs, International Journal of Computer Science and Network Security (IJCSNS) 17 (3) (2017) 38.

[8] I. Gorton, J. Klein, Distribution, data, deployment, STC 2015 (2015) 78.

[9] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, M. Bone, The concept of reference architectures, Systems Engineering 13 (1) (2010) 14–27.

[10] M. Derras, L. Deruelle, J.-M. Douin, N. Levy, F. Losavio, Y. Pollet, V. Reiner, Reference architecture design: A practical approach, in: ICSOFT, pp. 633–640.

[11] I. Iso, Information technology — reference architecture for service oriented architecture (soa ra) — part 1: Terminology and concepts for soa, International Organization for Standardization (2016) 51. URL https://www.iso.org/standard/63104.html

[12] G. Muller, A reference architecture primer, Eindhoven Univ. of Techn., Eindhoven, White paper (2008).

[13] L. Bass, I. Weber, L. Zhu, DevOps: A software architect's perspective, Addison-Wesley Professional, 2015.

[14] S. Angelov, P. Grefen, D. Greefhorst, A classification of software reference architectures: Analyzing their success and effectiveness, in: 2009

Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, IEEE, 2009, pp. 141–150.

[15] M. J. Page, D. Moher, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, et al., Prisma 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews, Bmj 372 (2021).

[16] M. L. Rethlefsen, S. Kirtley, S. Waffenschmidt, A. P. Ayala, D. Moher, M. J. Page, J. B. Koffel, Prisma-s: an extension to the prisma statement for reporting literature searches in systematic reviews, Systematic reviews 10 (1) (2021) 1–19.

[17] B. A. Kitchenham, D. Budgen, P. Brereton, Evidence-based software engineering and systematic reviews, Vol. 4, CRC press, 2015.

[18] M. Borrego, M. J. Foster, J. E. Froyd, Systematic literature reviews in engineering education and other developing interdisciplinary fields, Journal of Engineering Education 103 (1) (2014) 45–76.