IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2018, accepted September 15, 2018. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.2872673

Framing a Sustainable Architecture for Data Analytics Systems: An Exploratory Study

JITONG ZHAO, YAN LIU, AND PENG ZHOU

AQ:1 Tongji University, Shanghai 200092, China

AQ:2 Corresponding author: Yan Liu (yanliu.sse@tongji.edu.cn)

AQ:3 This work was supported by the National Key Research and Development Program of China under Grant 2018YFC0704600. The work of J. Zhao was supported by the China Scholarship Council under Grant 201706260040.

AQ:4 **ABSTRACT** Data analytics systems (DASs) with big data capabilities have started playing a promising role in online service ecosystems and large-scaled interconnected systems of many enterprises. The rapid development of analytics models and technologies, along with affordable infrastructures and accumulated data repositories, leads to encouraging expectations on DAS, while also bringing challenges in terms of how to deal with the increased development complexity. However, systematic methodologies for designing a sustainable DAS are still missing. To harness the dynamics raised by technology evolution, ambiguous requirements, under-explored data environments, and so on, framing a sustainable software architecture turns out to be a critical task. By exploring the complex nature of DAS, we propose a novel approach, sustainable architecture development for DAS (SstAD-DAS), to provide practical guidelines for architecture development. A shock absorber mechanism is presented to harness the dynamics of DAS and facilitate the development of a sustainable architecture, the "long decision chain" challenges are handled with a generic process model, and collaborations and responsibilities of participants are suggested to enable better model implementation. SstAD-DAS allows architects to accommodate the long decision chain, leverage skill sets from multiple contributors, and evaluate architectural decisions continuously. Finally, this paper demonstrates the capability and usability of SstAD-DAS by sharing experiences and observations from the continuous development of an intelligence analysis system.

INDEX TERMS Architecture sustainability, BPMN, data analytics system, architecture development method.

I. INTRODUCTION

With the continuous development of information technology and the increasing maturity of database technology, the amount of data is growing rapidly. This development calls for data analytics systems that are capable of dealing with massive amounts of data and for obtaining valuable information through in-depth analysis [1]. Data analytics systems (DAS) are built utilizing data mining, predictive analytics, and machine learning tools in order to discover useful information, suggest conclusions, and support decision-making [2]–[4]. DAS cover whole data processing chain, including collecting, inspecting, cleansing, transforming, modeling, and visualizing data. They are widely used in commercial industries and help organizations draw conclusions about various kinds of information they contain [5]. Meanwhile, the increasing usage of data analytic applications and the complexity of data types have sparked the technological development of Big Data and Cloud Computing [6]. The enrichment of these data analytics techniques and plenty of existing information data all promote the rapid development of data analytics systems.

A considerable number of data analytics systems have been applied extensively in various domains, such as climatology, logistics, and intelligence [7]. This kind of system integrates domain knowledge, data analysis knowledge, and software architecture knowledge; in all of these, architectural design plays a significant role. A range of research work has been carried out on detailed architecture development. The social data analysis systems presented by Lee *et al.* [8] provide social data gathering agent and analysis modules to conduct semantic analysis of large-scale Social Network Service (SNS) data. The healthcare cloud system proposed by He and Zhao [9] constructs a reasonable architecture for efficient and unified healthcare data storage, analysis, and

management. An online global climatic data system [10] provides a rational model of World Meteorological Organization (WMO) climate data and demonstrates functions of web-based data sharing, search, visualization, and analysis. These systems are tightly coupled with detailed application scenarios. They bring inspiration to system development in specific domain, providing useful help and references. However, no general approach exists that provides design guidance for building a data analytics system.

A more mature approach for architecting DAS is needed, as there is a huge and growing gap between what end-users expect from their data analytics systems and what they actually get [11]. Currently, many data analytics systems collect massive amounts of data and store them in a Hadoop environment based on Hbase or HDFS, such as financial Big Data and human resources Big Data [12]. Nosql databases are also introduced and the data warehouse technologies are utilized to enhance data analytics systems recent years [13]. However, information collection and data analysis go far beyond piling up numerous data and putting them together, which is far from reaching their goals. Well-designed DAS are expected to help users make more informed decisions through data mining, predictive analytics, machine learning, leveraging domain knowledge and so on.

In order to enhance the ability of data analysis in practical application, we need to understand the potential challenges and to create a complete methodology for DAS development. Architecture development method is an integral part of building up an efficient, scalable, and long-term reliable DAS. System architecture decisions are usually made in early phases of the software development, and iteratively enhanced during the overall development lifecycle [14]. Therefore, the architecture has a vital influence on how requirements are analyzed and what kind of technology solution the construction takes. It is the first step and the most basic part in the development methodology. Architects must create designs that can endure throughout the evolution of the software. To achieve sustainable architectures, we need sustainable design decisions, and the cost efficiency of required changes to those decisions [15].

In this paper, our study focus on architecture development method for building DAS. To make architectural design decisions successful, it is important to take into account the key characteristics of DAS. DAS usually involve many complexities, which entail challenges for the construction of a sustainable DAS architecture. Tough data analytics tasks, the need for diverse analytics skill sets, and the complex data-centric ecosystem all have an influence on decision-making during the development of a DAS architecture. Successful sustainable DAS architectures are reported by today's Internet giants, such as Google,[1] Netflix,[2] and LinkedIn.[3] They have shared their experience on architectural decision-making and publish their reference architecture on their official websites.[4] However, these solutions are all tightly coupled with business logic, and difficult to clone and apply in other situations for reasons related to the data itself, to funds, and to human resources. Thus, there exists no common practice for establishing a sustainable architecture for DAS.

To explore how to build a sustainable architecture for DAS, we conducted a field study to help understand the fundamental technologies and the specifics of the DAS. Considering the long decision chain of DAS, we thoroughly studied the challenges of decision-making associated with it. We attempted to overcome these challenges for building a DAS architecture in our study. An approach called SstAD-DAS was proposed and a preliminary process model for DAS architecting was presented to provide a reference for other practitioners. In addition, we used a concrete case study to demonstrate how to use this approach. This case study also implies that our approach can help adapt to the challenges and establish a sustainable DAS architecture.

The key contributions of this paper include:

1) Analyzing the challenges of building a DAS architecture, which include essential complexity, system dynamics, and the long decision chain of DAS.
2) Proposing an architectural approach called SstAD-DAS (Sustainable Architecture Development for Data Analytics System), which considers the perspectives of different stakeholders; presenting a process model, which provides guideline for DAS practitioners, and covers the complicated decision-making.
3) Bringing a case study for sharing practical experiences in using SstAD-DAS to establish a sustainable DAS architecture.

The remainder of this paper is organized as follows. Section II describes the related work and Section III introduces the process of our exploratory study. Section IV summarizes specific architectural challenges of building DAS. Then Section V proposes a novel approach called SstAD-DAS for framing a sustainable data analytics architecture. Section VI shows a case study sharing our experience in designing a DAS architecture using SstAD-DAS and Section VII concludes this paper.

II. RELATED WORK

Different aspects related to software architecture and data analytics systems have gained considerable research attention in recent years. This section presents existing contributions and is organized into the following categories: We will first introduce advantages of software engineering for data analytics systems. Next, we will share some related work about sustainable software architectures. Finally, some classical cases of software architecture for data analytics systems will be analyzed briefly.

[1] http://www.google.com/
[2] https://www.netflix.com/
[3] https://www.linkedin.com/
[4] https://medium.com/netflix-techblog/system-architectures-for-personalization-and-recommendation-e081aa94b5d8

A. SOFTWARE ENGINEERING FOR DATA ANALYTICS SYSTEMS

Nowadays, Big Data is permeating numerous aspects of human life, in particular in the data analytics domain. Research regarding software engineering for data analytics systems has aroused extensive interest in various studies [16]–[18]. In these works, the growing role of cloud computing in Big Data ecosystems has been studied and the establishment of appropriate infrastructures has been introduced, such as scalable data warehouse architectures and integrated storage systems for Big Data analytics. In addition, a considerable number of data analytics techniques have been developed from various perspectives in order to develop data analytics systems systematically [19], [20]. The research presented in [21] summarizes the characteristics of all the major single-case analytical techniques and provides a set of recommendations for choosing appropriate data analytics techniques suitable for different situations. Other contributions, e.g. [22], [23], present detailed architectural solution blueprints especially for designing and establishing Big Data analytics services across enterprises by utilizing common design components and practical standards.

B. SUSTAINABLE SOFTWARE ARCHITECTURE

In recent years, a great amount of research work has also been carried out on improving the understanding of sustainable architectures to provide support during the architectural development process. Naab *et al.* [24] emphasized that data needs more attention in sustainable architecture development and shared experiences from prototyping a large-scale application ecosystem. Sherman and Hadar [25] identified the need for a sustainable architecture maintenance process and proposed a solution for motivating professional architects to maintain architectural documents. Zdun *et al.* [15] presented several criteria to help architects assess the sustainability of their architectural design decisions and offered the lessons learned in their work as guidelines for achieving sustainable decisions.

Furthermore, in a wide range of work, interests are expressed in numerous development approaches for sustainable architecture [26], [27]. We have completed a technical report of a sustainable intelligence analysis system architecture in our early work. In this paper, we have proposed a practical guideline for framing a sustainable architecture for a specific domain, namely the data analytics domain, considering the long decision chain throughout the overall architecture development lifecycle.

C. SOFTWARE ARCHITECTURE OF BIG DATA SYSTEMS

Li *et al.* [5] and Demchenko *et al.* [28] investigated how to improve current comprehension of Big Data architectures to provide good decision-making for the construction of data analytics systems. Architectural components, key technologies, and measures for Big Data are defined in research, and suggestions for addressing data challenges in architecture are presented.

Meanwhile, a wide variety of heterogeneous architectures technologies have been applied for the implementation of data analytics scenarios [11]. The publications and classical architecture definitions have mainly focused on framing architectures of proprietary solutions, including Twitter [29], LinkedIn [30], [31], and Facebook [32] in the social network application domain. Other data analytics cases such as Netflix [33] (capturing value from commercial video-streaming), and BlockMon [34] (monitoring network traffic through a high-performance analytics platform) have also been demonstrated by researchers. Moreover, from another perspective, work in [35] and [36] focuses on infrastructure, such as the features and hardware of data center networks, by using qualitative and quantitative analysis.

III. EXPLORATORY ROADMAP

There has not been systematical research for DAS architecture development method, thus we did some exploratory study. As shown in Figure 1, this section presents our roadmap of the exploratory study. Before embarking on research on how to realize a sustainable system architecture, a deep understanding of DAS is needed. We studied the challenges of building DAS from different perspectives, and summarized the problems found in the design process, including essential complexity, development dynamics, and the long decision chain.

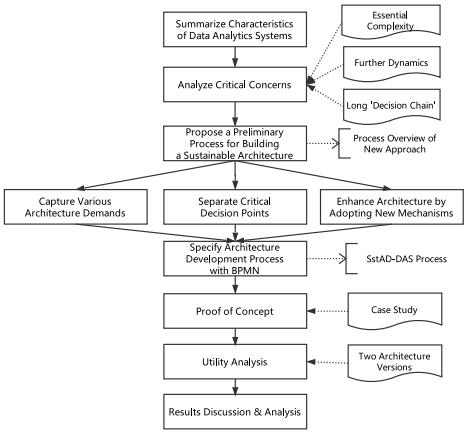


FIGURE 1. Process of the exploratory study.

We proposed SstAD-DAS, a preliminary approach to build a sustainable architecture, which was subsequently refined by capturing various demands, summarizing our findings regarding critical decision points and adopting some new mechanisms. Then, a process model was provided to specify the overall architectural development process and help guide how to use SstAD-DAS. Finally, we shared a case study on the architecture of an intelligence analysis system. Through our comparison study between two architecture versions, we intended to prove the architectural availability and analyze the results at last.

IV. UNDERSTANDING THE CHALLENGES OF BUILDING A DATA ANALYTICS SYSTEM

During the development process of DAS, stakeholders such as data engineers, data modelers and so on usually encounter a lot of difficulties. However, not many systematic studies of such challenges and related concerns can be found in the research literature. To better understand and design DAS, this section analyzed the characteristics of DAS from the perspective of complexity, dynamics, and long decision chain.

A. ESSENTIAL COMPLEXITY OF ARCHITECTING DATA ANALYTICS SYSTEMS

According to the characteristics of DAS, the essential complexity of DAS can be organized in three aspects:

(1) The continuous evolvement of data ecosystem needs to be taken into account when making architecture decisions. Nowadays, various fields have begun to develop their own data ecosystems, which provide proper data model to middleware, REST API, Web IDE and so on and accomplish data intelligence analysis. Data ecosystems take charge of various data tasks, for example, Apache Hive[5] supports highly efficient data storage and data analysis of large datasets stored in Hadoop. Apache Spark[6] acts as a unified analytics engine for large-scale data processing. Diverse skill sets are required for understanding the data-centric ecosystem and processing different data types, data storage methods, data mining approaches and so on.

(2) Extra modules, components, services, and mechanisms should be designed to meet the growing data requirements and to avoid data loss or permanent data corruption [37]. Storing and preprocessing multi-source heterogeneous data also increases the complexity of DAS. An integration technique should be designed to enable preprocessing across different heterogeneous datasets (unstructured, semi-structured, and structured) [1]. In addition, data preparation can help identify significant data, format it appropriately, and generate a smaller dataset with better quality, which can significantly improve the efficiency of the modeling phase [38]. Pre-computing should also be considered to deal with intermediate results at the beginning of algorithms' execution to fasten the calculation and to enhance the efficiency of system substantially.

(3) A well-designed DAS system is highly dependent on data visualization, statistical analysis, and data mining technologies, and it involves various stakeholders, such as domain expert, data scientist, end-user and so on. Various statistical analysis [39] and data mining approaches [40] have been developed for the purpose of summarizing data, drawing inferences, and discovering accurate information. In order to facilitate the analysis and help people understand information effectively and quickly, the analysis results and the statistical reports are visualized using visualization approaches [41] such as Vis.js,[7] D3.js.[8]

B. DYNAMICS OF DATA ANALYTICS SYSTEMS DEVELOPMENT

In addition to the essential complexity of DAS, the dynamics of DAS is also a key point for understanding DAS. Requirements changes and technology upgrades are unavoidable during the architecture development process and further increases the complexity of such systems.

The dynamics of DAS includes two aspects. To begin with, the requirements in data analytics systems are highly dynamic. Unlike in traditional information systems, there is no strict distinction between architecture development and requirements. In many cases, developers realize that their data analytics requirements are fuzzy and that iterative refinements of the requirements are required as development progresses [42]. It often happens that an architect progressively explores functionalities that the designed architecture could support and that new data analytics requirements are determined [43].

Furthermore, emerging technologies also make DAS highly dynamic, which include the springing up open source products, promising commercial one stop solutions, rising architecture patterns such as lambda [44], and increasingly important development methodology such as DevOps [45]. Realizing the target architecture is a difficult issue due to the great variety of technologies and skill sets [1]. Who should be responsible for these tasks is also not clear, such as whether we need on-site data engineer or outsourcing to deal with the complex data and so on. Worse still, there are always frequent changes regarding the stakeholders, which makes developers confused about the current user scenarios. All of this should be taken into account when framing a sustainable architecture for DAS [46].

C. DIFFICULTIES OF HANDLING THE LONG DECISION CHAIN

The architectural decision chain is the set of decisions that architects face while framing a sustainable DAS architecture. It illustrates the decision-making process [46] in software architecture.

In contrast to traditional information systems, whose decision chain is short and explicit, data analytics systems possess a long decision chain, along with a complex data value chain. However, there is no systematic research on questions such as which stakeholders are involved in the decision chain, who should be responsible for which task, how to accomplish the tasks and so on. As illustrated in Table 1, there are numerous multi-disciplinary stakeholders involved in the chain. It is the multitude of stakeholders that makes this such a complex process. Even if a stakeholder has the final say on various decisions, determining the long decision chain for the

[5]https://hive.apache.org/
[6]http://spark.apache.org/
[7]http://visjs.org/
[8]https://d3js.org/

TABLE 1. Stakeholders involved in architecture solutions.

Stakeholders	Description
User	Represents consumers of data analytics systems to whom the architectural solution seeks to offer accurate information.
Operator	In the general sense, these groups of users are in charge of daily operation and maintenance work of data analytics systems.
Domain Expert	Represents specialists familiar with some specific domain who specify all scenarios for a particular transaction and identify possible requirements.
Product Manager	Product managers cooperate with the development team, communicating with users and taking responsibility for the functionalities of the product system.
Designer	The designer focuses on the comprehension of the users' requirements, the design of the preliminary product mock up, and the visualization of the display prototypes during the prototype design phase.
Prototyper	Represents developers who design the software prototype system and help support the architectural solution through an analysis of the key techniques and a proof of concept.
Architect	Architects focus on framing a sustainable architecture solution for a data analytics system and play an important role in the overall software development lifecycle.
Ops	Software operations (Ops) is a part of DevOps; its members are responsible for the integration of the production environment, the maintenance of current products, the release of new products, including system administrators, network technicians, and database administrators.
Developer (front-end &back-end)	Represents engineers who are responsible for the primary development of the overall system architecture, including front-end programming and back-end programming.
Data Analyst/ Engineer	Represents engineers who are responsible for the analysis of data sources from various information systems, the processing of massive amounts of data, and the development of ETL scripts.
Data Consumer	Represents the architecture solution participator who consumes large amounts of information in a bid to deliver data analysis solutions.
Data Provider	Represents existing systems and provides basic information from diverse data. Examples: airline systems, hotel systems, Internet cafe systems.
Data Scientist	Data scientists evaluate data analysis approaches and help make sense of the massive streams of digital information they glean from the system.
Visualization Expert	By applying visualization techniques, visualization experts aim to improve the understandability of the data model and strengthen the reliability of the analysis results.
Data Modeler	Data modelers are responsible for both logical and physical warehouse data modeling, including data structure design and model validation.

architecture is still a complex process. For instance, a fairly long time is required to obtain sample data from the complex data environment and evaluate the data models based on the experiment results. How to capture architectural demands and how to overcome the long decision chain during architecture evolution are the key research questions of this paper.

Compared with information systems, which usually focus on business logic, DAS focus more attention on data storage, data algorithm analysis, data model design, and other data analytics aspects. Thus, domain experts, data engineers, data consumers, data scientists, data modelers, and visualization experts all participate in the long decision chain. For example, users might not be able to describe their expectations clearly, since they might have no idea about what kind of accurate information current technologies can help them mine from miscellaneous data. In such situations, data engineers will first explore the available data sources and investigate the data environment. Data scientists will then research analytical models and visualization experts will demonstrate the feasibility. Domain experts, product managers, and users will work together to determine the requirements based on their research using a data analytical model.

V. SstAD-DAS: A PRELIMINARY ARCHITECTING METHOD FOR DAS

Aiming at handling the challenges described above, we proposed an architectural construction approach called SstAD-DAS (Sustainable Architecture Development for Data Analytics System). It draws on the existing architecture development methods, and combines the characteristics of DAS. Firstly, we introduced its fundamental process, which includes four main phases and their intermediate artifacts. Data exploration report, the technical architecture, architectural mechanisms specialized for DAS, and final architectural solution are four deliveries conducted in the fundamental process. Subsequently, we provided design principles to help accomplish the key phases and produce corresponding deliveries. Finally, considering the perspectives of all the stakeholders, we presented a detailed architectural design process, and modeled it by BPMN. It provides practical guidelines for professional architect teams to frame a DAS architecture. Practitioners can tailor this detailed process model according to their own project characteristics. Architects can also use this model as a reference to build commercial domain solution for data analysis.

A. SstAD-DAS PROCESS OVERVIEW

Figure 2 presents essential architectural tasks and depicts the fundamental process of the approach. When capturing architectural demands, architects should pay special attention to data-related issues and achieve a data exploration architecture. Subsequently, the architectural concerns should be separated based on domain knowledge, tasks, skill sets and stakeholders. In this way, possible overlaps of functionalities or requirements can be reduced and a technical architecture is completed. Then the architects need to accommodate the dynamics of the DAS and apply some mechanisms to harness it. Finally, a continuous architecture design is created and a sustainable architecture solution for DAS is obtained.

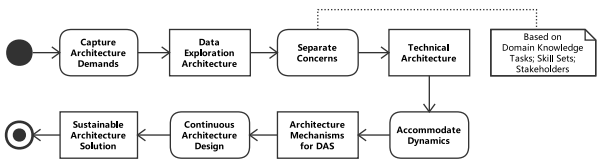


FIGURE 2. Process overview of SstAD-DAS.

1) ARCHITECTURAL DEMANDS

Capturing architectural demands in the preliminary design stage is important, since these are the foundation of the entire software system. Compared to traditional architecture, the data characteristics should be separated when analyzing the architectural demands of DAS. Different data sources, different levels of data quality, different levels of data frequency, and different domain knowledge accumulated from data all lead to different architectural demands and solutions.

Figure 3 shows the data-specific issues regarding DAS architectural demands. In addition to traditional architectural concerns [47], this section captures the demands from the perspective of data complexity and interprets considerations on functional requirements and non-functional requirements. The demands and considerations all surround the goals aiming at data analysis.

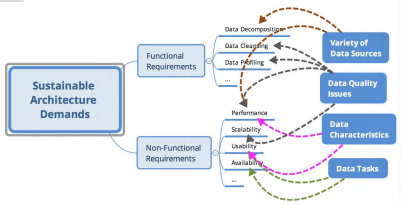


FIGURE 3. Data-specific issues for architecture demands.

If the target DAS consist of various previous systems and massive amounts of disordered data, architects usually pay a lot of attention to the complex data and the known-unknown legacy systems. The variety of data sources should be decomposed and refactored to ensure the performance of the architecture. Data profiling is a systematic analysis of the content of a data source [48], which helps architects to thoroughly and quickly unveil the content and structure of data. It can help ensure architectural performance and scalability by identifying data quality issues.

The data characteristics should be studied before the data modeling process is initiated, including amount of data, data format, required data processing time, character of domain data and so on. Architects design an architecture based on these characteristics. For example, architects have to decide whether to choose streaming processing or batch processing, depending on the processing time requirements. The data characteristics determine the selection of the architecture paradigm and affect architectural performance and usability. Data tasks in DAS emphasize the domain knowledge, storage, analysis, statistics, and visualization of enormous amounts of data, which is of vital importance for the usability and availability of the architecture.

2) SEPARATION OF CONCERNS

The next step is to perform a separation of concerns, i.e., to break down the architecture into distinct parts to make sure there will be as little overlap in functionality and requirements as possible. When separating the concerns of DAS, three key factors need to be considered, they are different tasks, skill sets, and stakeholders.

We handled the challenges of long architectural decision chain by modeling system demands (Figure 3), and divided the system architecture into independent modules based on different tasks and collaboration events. This ensures that the architecture is capable of accommodating special issues or unexpected circumstances.

Different skill sets and domain-specific tools are also key factors affecting the architectural design. Complex DAS usually involve various skill sets, which are associated with domain knowledge and require understanding of related fields. It is impossible even for an experienced developer or architect to handle the whole architecture alone. In such cases, even if two modules in combination with each other may produce a better effect, the architects always need to separate and decouple them from each other based on the different skill sets of the development team.

The stakeholders usually have a significant impact on the architectural decision-making process. Capturing and separating their interactions and collaboration is crucial for ensuring the success and sustainability of an architectural solution, especially for DAS.

3) SHOCK ABSORBER MECHANISM

A sound and sustainable architecture should be able to harness the dynamics of a specific domain. It is worth noting that our research is focused on a specific domain, namely data analytics systems. Data status, analysis model, data model, and requirement goals of DAS are often different from ordinary information systems. Mechanisms for this domain are far beyond the scope of classical design mechanisms. Consequently, specialized architectural mechanisms are needed, and more effort should be put into completely understanding the dynamics of data analytics. One of these important mechanisms is a shock absorber mechanism, whose design is a crucial task during the development of DAS architecture.

A data value chain in DAS aims at drawing value from strategic data assets, discovering information from data lakes, and transforming data step by step, to the final exposed terminal, throughout the whole data analytics system. Figure 4 explains the inner workings of data value chain of data analytics system. The data value chain[9] shown above the red line in illustration is taken from slides by Edd Dumbill (Silicon Valley Data Science) in Strata's big data conference.

Since not all the links of the data value chain are highly coupled with each other, the concept of a 'shock absorber' is introduced. A shock absorber is required between two adjacent steps. This concept can also be summarized as a damping architecture pattern especially devised for DAS. To explain mechanism of shock absorber more clear, illustration in figure 4 below the red line is modified by the author, adding shock absorber module (showed in the dotted rectangle).

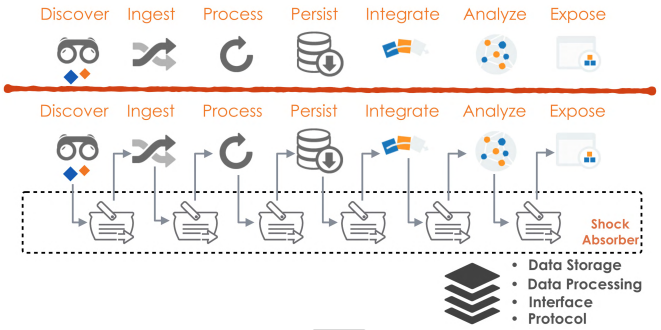


FIGURE 4. Shock absorber for data value chain.

A shock absorber is not merely a simple middleware or interface. DAS are not just systems with input and output; the data itself needs some buffer zone. For instance, data sets stored in the Integrate Step have a rather huge data volume. The Analyze Step performs analysis on data models with different levels and from different perspectives. The shock absorber between the Integrate Step and the Analyze Step has to process the massive amounts of data with different structures and characteristics. Thus, a processing component is required in the shock absorber in addition to an interface component.

Take for another example, faced with the uncertainty of analysis data, we neither manipulate the production data directly nor establish a data channel. Manipulating data in production environments can result in numerous problems, including down-time for the application. These can impact stability, and client perception. The common approach is to extract, transform, and load such data into a temporary data environment, which can be thought of as a shock absorber. The concrete implementation form of shock absorber is associated with the layer it is in and designed depending on the specific circumstances. In addition, more and more open source and commercial applications are playing the role of building shock absorbers, such as Kettle.[10]

4) CONTINUOUS ARCHITECTURE DEVELOPMENT

Nowadays, software development processes are undergoing huge changes, including architecture design and large-scale development. The architecture consists of two parts, the architecture design and the architectural Proof of Concept[11] (PoC). PoC is a realization of a certain method or idea in order to demonstrate its feasibility, but does not represent a deliverable.

[9] http://conferences.oreilly.com/strata/big-data-conference-uk-2015/public/schedule/detail/39796

[10] https://community.hitachivantara.com/docs/DOC-1009855

[11] https://www.techopedia.com/definition/4066/proof-of-concept-poc

That is to say, there is a team of professionals designing the data analytics architecture, not just architects. Architectural developers also participate in the design of the overall architecture by offering rapid prototyping and developing a PoC. After the architecture team has succeeded in achieving the goals of this PoC, large-scale development can start, which again involves its own designers and programmers. The relationship between the architecture team and the large-scale development team is bidirectional. They stimulate one another and form an overall architectural lifecycle.

B. A DETAILED PROCESS OF SstAD-DAS

To gain a deeper understanding of SstAD-DAS, a detailed process model was created as an essential process analysis model. We chose BPMN to model this architectural development process. As shown in Figure 5, this process model is provided as a reference implementation solution of SstAD-DAS and is able to guide architecture teams in developing a sustainable DAS architecture. This detailed process is able to handle the long decision chain of DAS. Considering the perspectives of all the stakeholders, we separated the process into different modules and defined detailed tasks for each stakeholder. They take right responsibilities based on what they are good at. The process figure can also be browsed on the Internet by url 'https://www.processon.com/view/link/5b471fcde4b054aa54b401ae'.

The Requirements Elicitation and Product Delivery Pool is responsible for the activities closely related to users and specific domains outside the developers' control. This pool should be divided into two lanes. The Requirements Elicitation Lane is in charge of requirements elicitation and iterative prototyping, while the Product Delivery Lane is in charge of deployment and delivery, as well as product acceptance. In order to simplify the discussion, we put both of these lanes into this pool. Requirements changes are described as Intermediate Events in this pool, and are followed in this subprocess by tasks such as clarifying new requirements and analyzing potential solutions, updating product mockups, and incremental prototyping.

After completion of the requirements elicitation phase, the requirements are sent to the data environment investigation step via a message flow. It is worth noting that in the DAS developing process, the information about the research on the data analytics model is sent back to the Iterative Prototyping sub-process. Afterwards, the requirements are re-examined by multiple stakeholders and the prototype design is refined. The requirement prototype and refined data analytical model are sent to Solution Development Process to help large-scale system development. After continuously DevOps, final system is deployed, reviewed, and accepted.

The Solution Development and Architecture Evolvement Pool focus on various implementation technologies and encompasses development mechanism research, proof of concept, continuous system development, and completion of essential solutions. Adoption of new technical solutions is modeled as an Intermediate Event in this pool, and is followed in this sub-process by tasks such as prototyping & evaluation, architecture improvement, and upgrade of functionalities.

Meanwhile, the architects also should pay more attention to DevOps Pool, which encompasses numerous activities streamlining the software delivery process, improving the cycle time and emphasizing learning by streaming feedback from production to development [45]. The Continuous DevOps Pool consists of the Dev sub-process and the Ops sub-process, which deal with emerging technologies and multi-disciplinary skill sets. Nowadays, architecture design is not only related to the architecture itself, but also highly coupled to other development methodologies such as DevOps.

VI. CASE STUDY

We explored how to frame a sustainable architecture for DAS, presented a process model for it, and introduced our approach SstAD-DAS in this paper. However, the process model may be too abstract for people without enough experiences in complex DAS development. In order to make it easier for the readers to understand how to use our approach, we applied it into the development process of an IAGraph product, and finally, we found that using SstAD-DAS to build a sustainable architecture was promising.

A. IMPLEMENTATION BASED ON SstAD-DAS

We developed an intelligence analysis system called IAGraph, which was guided by our approach. Due to non-disclosure agreements, we have simplified some contents in our case study. This system involved numerous multi-disciplinary stakeholders. Its first vision lasted for six months, and second vision lasted for eight months. As it was an intelligence analysis system, parts of the screenshots had been obfuscated.

1) BACKGROUND FOR IAGraph

IAGraph is a complex data analytics system, which was launched in 2015 for intelligence analysis. It aims to analyze criminal information and help identify suspicious behaviors based on a variety of data from various information systems. For instance, we can analyze personal path information extracted from train and airline systems to detect similar suspicious paths and identify suspects. In our project, we stored the processed data in the graph database Neo4j, which is suitable for the storage and analytics of social networks.

Figure 6 shows four screenshots depicting the effectiveness of this DAS. In this project, the user language is Chinese. However, it is still possible to identify the characteristics of DAS, including graph visualization, data filter, and data analysis. Figure 6(a) and Figure 6(b) illustrate the relationships among a group of suspects in a graph structure. Figure 6(c) exhibits the detailed filter conditions. Figure 6(d) displays all the suspects having a relationship with a specific single person. After the necessary RESTful interfaces had been designed and a flexible system had been developed, we were able to obtain the results of Figure 6(d) by using the form

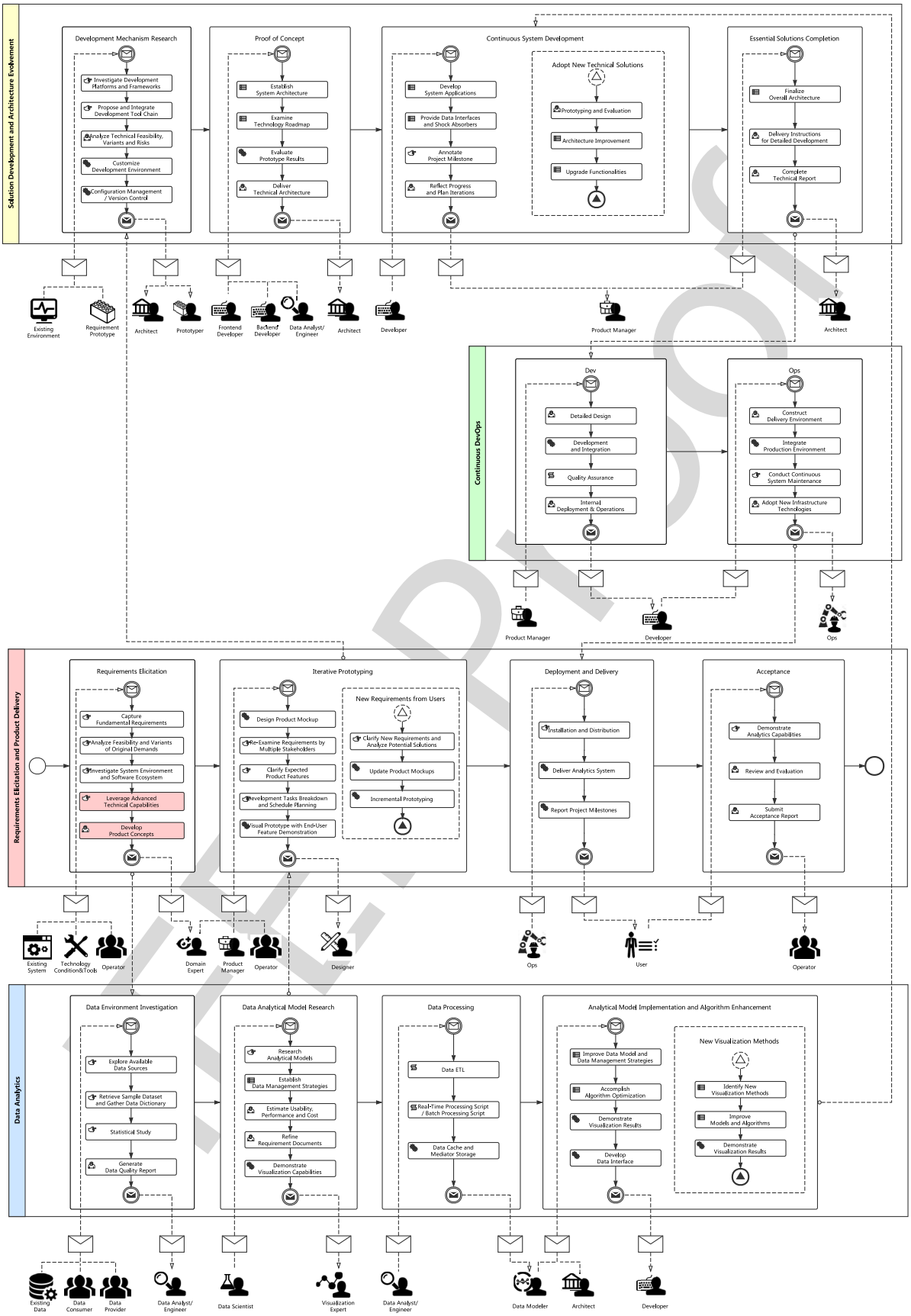


FIGURE 5. SstAD-DAS process.

FIGURE 6. IAGraph system screenshots.

of 'ip:port://relations/ normal/identification_card_number'. The returned json data example is presented in Figure 7(c).

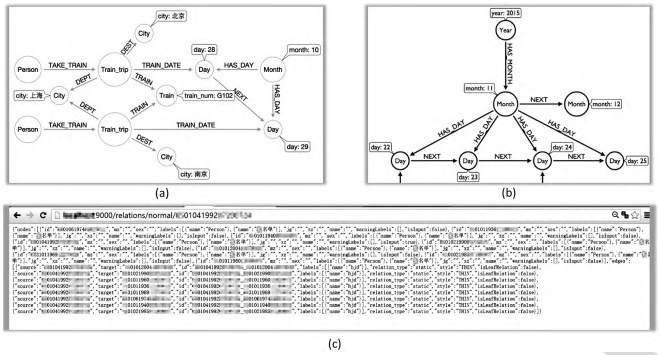


FIGURE 7. Graph data structure designed for system.

2) DATA EXPLORATION FOR IAGraph

According to the process model, investigation of the architecture demands is the first step. Before the data analysis can be conducted, the data scientist must know what variables are included, how many cases are in the dataset, and how many observations are missing.

An exploration of the data helps to answer these questions. Thus, in our case study the data engineers explored relevant data in various information systems, which were stored in Oracle database. These information included personal basic profiles, communication data, traveling data, criminal information data, and transaction data. The single-table data often reaches hundreds of millions of records, and records will increase with time. Traveling data increase per day and criminal information increase in real time, which determine the way of incremental data synchronization. These data are all structural so that we could operate them in Relational database such as Postgresql. There are several missing value in traveling data, so proper ETL should be performed.

3) SEPARATION OF CONCERNS FOR IAGraph

As proposed by the SstAD-DAS, specific concerns should be separated. This architecture is designed to be divided into decoupled modules. Modules such as algorithm implementation, data storage approach, front-end development, DevOps, and so on are all separated from each other to allow realization by specialized professionals with different skill sets or using different domain-specific tools.

For example, the availability of new visualization methods generally has no effect on our data analytics system since the 'Algorithm Models Implementation' module (application layer) has been separated from the others. Moreover, in each module, data interfaces are reserved especially for further development and refinement.

4) SHOCK ABSORBER MECHANISM FOR IAGraph

Our shock absorber is a composite of four major components: data storage, data processing, interface, and protocol. Taking the shock absorber between the analysis of the results and the exposure of the data as an example, we designed a shock absorber to enhance the performance and effectiveness of the visualization. The information for discovering potential crime clues that was to be displayed was stored in a temporary NoSql database. Processing scripts programmed in jQuery[12] or other languages were required in the data visualization process. In addition, data interfaces were generally designed in advance, such as the data format of nodes, relationships, and so on. In addition, we utilized RESTful interactions[13] based on the HTTP protocol, which is pretty stable across different software product vendors. These tasks are all achieved by the four layers in the visualization shock absorber.

We also designed a shock absorber to manipulate the data from various information systems provided by involved data providers, such as personal basic profile, communication data, traveling data, and so on. The shock absorber extracts, transforms, and loads these data into a temporary data environment (graph databases in this architecture) to avoid manipulating data in production environments directly. There was also a shock absorber to process data stored in the graph databases before the analysis phase started. Some basic statistics were calculated in advance, such as information on personal relationships, to ensure optimal performance of our system.

5) IAGraph ARCHITECTURAL EVOLUTION

Figure 8 illustrates the preliminary design of the data analytics system architecture as of 30 April 2015. This architecture took our approach as guidance, sticking to the combination of theory and practice. The solid lines represent modules or functionalities realized in practice, while the dotted lines represent those that are reserved for the next version and that are merely interfaces at present. The actual system is very complex, so here we only demonstrate a simplified version of the software architecture, emphasizing key points.

Figure 9 presents the upgraded design of our data analytics system architecture as of 30 April 2016. The red lines represented modules or functionalities realized in the upgraded

[12]https://jquery.com/
[13]https://www.ibm.com/developerworks/library/ws-restful/

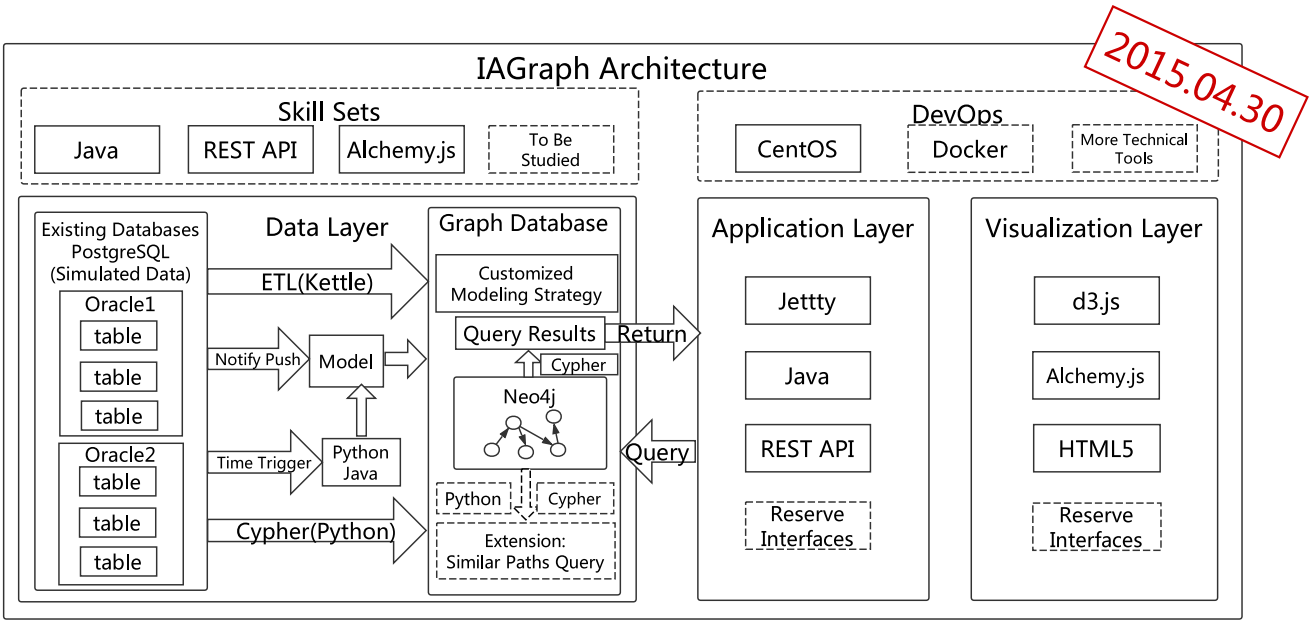


FIGURE 8. IAGraph architecture version I.

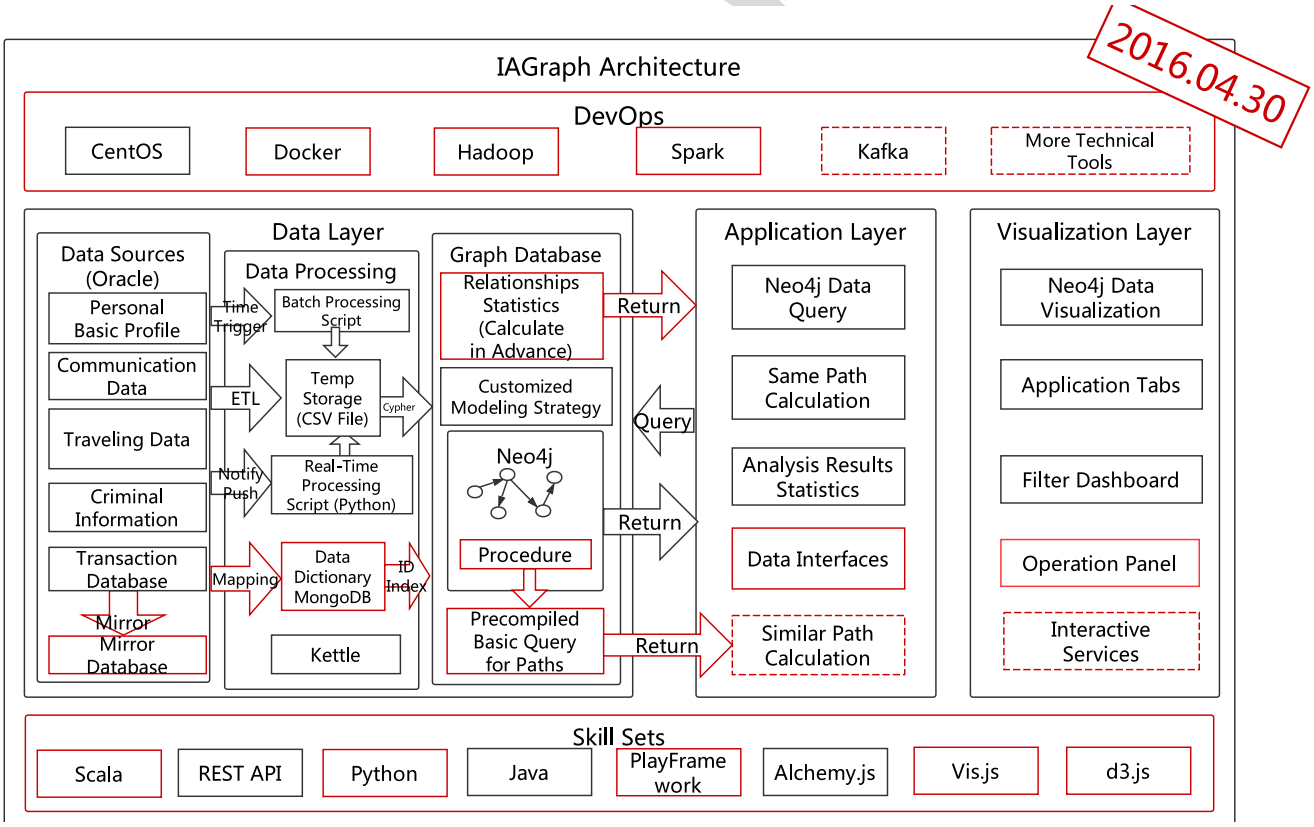


FIGURE 9. IAGraph architecture version II.

version, which had been reserved as interfaces in the previous one.

Since 2017, the system has been further migrated to the private cloud platform of host organization, leveraging some big data processing infrastructures. Both versions supported porting to Alibaba Cloud and meet the requirements of each stage, which proved our architecture was sustainable.

TABLE 2. Features improvements between the two versions.

System Feature	Version I	Version II
Data Processing Module	Reserved Interface	Data Dictionary
Relationship Statistics	Reserved Interface	Realized
Basic Query for Paths	Realized	Updated
Graph Data Storage Module	Realized	Refactored
Same Path Calculation	Reserved Interface	Realized
Similar Path Calculation	Reserved Interface	Reserved Interface
Data Visualization	Realized	Updated
Operation Panel Module	Reserved Interface	Realized
DevOps	CentOS & Docker	Continuously Tried New Tools

B. BUSINESS PROCESS MODEL FOR IAGraph

In order to thoroughly comprehend the sustainable architecture guided by our approach, a comparison study was performed between two different architecture versions during the architectural evolution.

According to the decision process model, requirement elicitation is the first step. In the Requirement Elicitation Pool, the intelligence analysts provided requirements, that they wanted to discover suspicious trails. There were many existing information system right now, which included personnel information management system, traveler information system and so on. Product Manager designed and modified the prototype iteratively based on the requirements from the police and data quality report from the data scientists.

In the Data Analytics Pool, the data engineers explored data environment as mentioned in section 5.1.2. Based on the generated quality report, the data scientists researched various analytical models and demonstrated different visualization capabilities. We choose graph structure to store and manage information due to the data characteristics. Take figure 7 for an example, figure 7(a) shows the relationships among people nodes and trip nodes, which store traveling information such as train traces, flight traces, hotel records, and so on. Figure 7(b) explains the principle of time tree split, where year, month, and day are designed as independent nodes. This enhances the query efficiency regarding information about the intervals between different trips.

Then the data engineers started to process related data, utilizing specialized tools such as Kettle to extract, transform, and load data from various data sources to the target graph database. Meanwhile, a batch processing script and a real-time processing script were programmed to help with incremental data synchronization. A customized modeling strategy was prepared by the data modeler to store the relationships between personal and detailed information. To facilitate querying suspects' common paths, the data architects considered making an extension for queries of each person's similar paths using Python/Cypher scripts.

In the Solution Development and Architecture Evolvement Pool, the developers established the primary IAGraph architecture and implemented the application layer and the visualization layer. After completing the PoC of this preliminary architecture, the developers began large-scale exploitation. Architects should consider the different skill sets and multi-disciplinary stakeholders in this phase and separate these concerns into decoupled modules.

The DevOps module in the IAGraph architecture (Figure 8) illustrates the tools we utilized, including CentOS, Docker, Jenkins and so on. After continuous DevOps, IAGraph system was deployed and delivered to the users.

C. DISCUSSION

In order to evaluate the availability of our approach, a comparison study was performed between two different architecture versions during the architectural evolution. Table 2 summarizes some important system features and subsystem improvements between the two versions.

In the primary version, some function modules could not be completed as development time was limited, and were left for development in later versions. Some new function modules could be added during the development of version II as the requirements had changed. These system features had all been designed as interfaces in the previous version, and could be developed and extended without any problem.

It is worth noting that the new version of the system architecture does not go beyond the previous one; it is merely an improvement or extension on the basis of the original one. The developers did not need to worry about anything being broken. Take the relationships statistics model, for example: This model was designed to be calculated in advance, which can greatly increase the query speed. In version I, this model was designed as a reserved interface; then it was completed in version II. As the architectural modules were separated and decoupled from each other, other modules were not affected by this change.

The system can also support changes and upgrades of technologies and tools. In the visualization layer of the upgraded version, we utilized Vis.js to replace D3.js. D3.js had been used in the previous version as the main instrument for visualizing the graph data due to the greater need for

maneuverability. We changed to Vis.js as it better suited our needs, including the styles and layout. As the system architecture was devised to separate modules from each other, changing the visualization submodule had no effect on the other modules, such as the data processing module or the common path querying module.

In the graph data storage module, we took into consideration extensions for common path queries. To resolve this issue, Python/Cypher scripts were prepared and executed in Version I. Neo4j was upgraded while we improved and refined Version II. The procedure, which is a form of extensibility, was one of the added features. All the query logic we had designed in the previous version could be reused and realized in a better way in the new one. This architecture has been confirmed to be sustainable, as it accommodates new releases of tools and technologies without affecting the development of the system and the infrastructure.

We applied our approach to frame an intelligence analysis system and investigated the availability of this approach. We set up the IAGraph architecture based on the SstAD-DAS, and provided the process model for it. The system was proven to be sustainable and allowed making changes between the two different versions. These changes did not affect other feature modules in our system. In this way, the effectiveness of the system development will continue to improve, benefiting from the sustainable architecture. Furthermore, the original architecture has not suffered any impact after our system was ported to big data infrastructure. In conclusion, our approach can support the evolution of the system.

VII. CONCLUSION

We analyzed the characteristics of data analytics systems from the perspective of complexity, dynamics and decision chain. The main purpose of this paper is to propose architecture guidelines for DAS based on practical experience. A process model was presented to help establish DAS architecture. In order to analyze the utility of our approach, we applied it into an intelligence analysis project. In this paper, we summarized practical experience in establishing this project as a case study.

Designing a sustainable architecture for DAS is a challenging task that cannot be resolved perfectly. What we have done is a trial procedure with no guarantees; success cannot be assured. There is not a single architectural design approach or recipe that can guarantee success. However, this is a good chance to improve our comprehension of the success factors. We want to share our practical experiences in order to stimulate more attentions to the demands on the architecture itself and offer practitioners some usable instructions on how to build large industrial data analytics systems and ecosystems. It is important to realize that DAS architecture has its own demands, and that DAS require a specialized design approach. Exploring the demands on the architecture empirically is a good research direction and will, hopefully, lead to a refinement of the architecture development process.

For future work, we intend to validate the sustainability of SstAD-DAS in subsequent versions of IAGraph and conduct more use cases with multi-disciplinary stakeholders to further refine SstAD-DAS.

REFERENCES

[1] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, Jul. 2014.

[2] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[3] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[4] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "CDAS: A crowdsourcing data analytics system," *Proc. VLDB Endowment*, vol. 5, no. 10, pp. 1040–1051, 2012.

[5] X. Li, F. Zhang, and Y. Wang, "Research on big data architecture, key technologies and its measures," in *Proc. IEEE Int. Conf. Dependable, Auto. Secure Comput.*, Dec. 2013, pp. 1–4.

[6] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, Jan. 2015.

[7] C. Elkan. (2013). *Predictive Analytics and Data Mining*. [Online]. Available: http://cseweb.ucsd.edu/~elkan/255/dm.pdf

[8] H. Lee, J. Her, and S. R. Kim, "Implementation of a large-scalable social data analysis system based on MapReduce," in *Proc. 1st ACIS/JNU Int. Conf. Comput., Netw., Syst. Ind. Eng.*, 2011, pp. 228–233.

[9] X. He and L. Zhao, "A data management and analysis system in healthcare cloud," in *Proc. Int. Conf. Service Sci. (ICSS)*, 2013, pp. 164–169.

[10] H. Xu and Y. Bai, "GCDViewer: An online data query, visualization and analysis system for global climatic data," in *Proc. Int. Conf. Agro-Geoinform.*, 2014, pp. 1–4.

[11] P. Pääkkönen and D. Pakkala, "Reference architecture and classification of technologies, products and services for big data systems," *Big Data Res.*, vol. 2, no. 4, pp. 166–186, 2015.

[12] P. C. Zikopoulos *et al.*, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. New York, NY, USA: McGraw-Hill, 2011.

[13] A. B. M. Moniruzzaman and S. A. Hossain. (Jun. 2013). "NoSQL database: New era of databases for big data analytics—Classification, characteristics and comparison." [Online]. Available: https://arxiv.org/abs/1307.0191

[14] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Reading, MA, USA: Addison-Wesley, 2012.

[15] U. Zdun, R. Capilla, H. Tran, and O. Zimmermann, "Sustainable architectural design decisions," *IEEE Softw.*, no. 6, pp. 46–53, Nov./Dec. 2013.

[16] J.-S. Kim, K.-Y. Whang, H.-Y. Kwon, and I.-Y. Song, "PARADISE: Big data analytics using the DBMS tightly integrated with the distributed file system," in *World Wide Web-Internet Web Information Systems*. 2014, pp. 1–24. AQ:6

[17] S. Sharma, "Expanded cloud plumes hiding big data ecosystem," *Future Gener. Comput. Syst.*, vol. 59, pp. 63–92, Jun. 2016.

[18] D. Linstedt and M. Olschimke, "Scalable data warehouse architecture," in *Data Vault 2.0*, D. Linstedt and M. Olschimke, Eds. Boston, MA, USA: Morgan Kaufmann, 2016, ch. 2, pp. 17–32. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128025109000027

[19] O. A. Nassar and N. A. Al Saiyd, "The integrating between Web usage mining and data mining techniques," in *Proc. IEEE 5th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Mar. 2013, pp. 243–247.

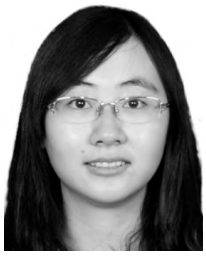
[20] M. K. Sohrabi and S. Akbari, "A comprehensive study on the effects of using data mining techniques to predict tie strength," *Comput. Hum. Behav.*, vol. 60, pp. 534–541, Jul. 2016.

[21] R. Manolov and M. Moeyaert, "Recommendations for choosing single-case data analytical techniques," *Behav. Therapy*, vol. 48, no. 1, pp. 97–114, 2016.

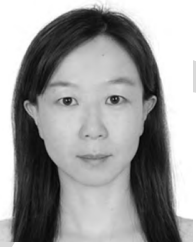
[22] P. Sarkar, "Big data and analytical services," Tech. Rep., 2015, p. 4. AQ:7

[23] M. M. U. Rathore, A. Paul, A. Ahmad, B.-W. Chen, B. Huang, and W. Ji, "Real-time big data analytical architecture for remote sensing application," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 4610–4621, Oct. 2015.

[24] M. Naab *et al.*, "Why data needs more attention in architecture design—Experiences from prototyping a large-scale mobile app ecosystem," in *Proc. 12th Work. IEEE/IFIP Conf. Softw. Archit. (WICSA)*, May 2015, pp. 75–84.
[25] S. Sherman and I. Hadar, "Identifying the need for a sustainable architecture maintenance process," in *Proc. IEEE 5th Int. Workshop Cooperat. Hum. Aspects Softw. Eng. (CHASE)*, Jun. 2012, pp. 132–134.
[26] H. A. Duran-Limon, C. A. Garcia-Rios, F. E. Castillo-Barrera, and R. Capilla, "An ontology-based product architecture derivation approach," *IEEE Trans. Softw. Eng.*, vol. 41, no. 12, pp. 1153–1168, Dec. 2015.
[27] E. Woods, "Aligning architecture work with agile teams," *IEEE Softw.*, vol. 32, no. 5, pp. 24–26, Sep./Oct. 2015.
[28] Y. Demchenko, C. de Laat, and P. Membrey, "Defining architecture components of the big data ecosystem," in *Proc. IEEE Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2014, pp. 104–112.
[29] M. Oussalah, F. Bhat, K. Challis, and T. Schnier, "A software architecture for Twitter collection, search and geolocation services," *Knowl.-Based Syst.*, vol. 37, pp. 105–120, Jan. 2013.
[30] R. Sumbaly, J. Kreps, and S. Shah, "The big data ecosystem at Linkedin," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 1125–1134.
[31] A. Auradkar *et al.*, "Data infrastructure at Linkedin," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Apr. 2012, pp. 1370–1381.
[32] H. Barrigas, D. Barrigas, M. Barata, J. Bernardino, and P. Furtado, "Scalability of facebook architecture," in *New Contributions in Information Systems and Technologies*. Cham, Switzerland: Springer, 2015, pp. 763–772.
[33] X. Amatriain and J. Basilico, "Recommender systems in industry: A netflix case study," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 385–419.
[34] D. Simoncelli, M. Dusi, F. Gringoli, and S. Niccolini, "Scaling out the performance of service monitoring applications with BlockMon," in *Passive and Active Measurement*. Berlin, Germany: Springer, 2013, pp. 253–255.
[35] T. Chen, X. Gao, and G. Chen, "The features, hardware, and architectures of data center networks: A survey," *J. Parallel Distrib. Comput.*, vol. 96, pp. 45–74, Oct. 2016.
[36] T. Singh and V. S. Darshan, "A modern data architecture with apache Hadoop," in *Proc. IEEE Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Oct. 2015, pp. 574–579.
[37] P. Vassiliadis, "RADAR: Radial applications' depiction around relations for data-centric ecosystems," in *Proc. IEEE 27th Int. Conf. Data Eng. Workshops (ICDEW)*, Apr. 2011, pp. 62–67.
[38] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Appl. Artif. Intell.*, vol. 17, nos. 5–6, pp. 375–381, 2003.
[39] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Hoboken, NJ, USA: Wiley, 2003.
[40] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.
[41] J. Kehrer and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 3, pp. 495–513, Mar. 2013.
[42] B. Nuseibeh, "Weaving together requirements and architectures," *Computer*, vol. 34, no. 3, pp. 115–119, Mar. 2001.
[43] R. Mohanani, P. Ralph, and B. Shreeve, "Requirements fixation," in *Proc. ACM 36th Int. Conf. Softw. Eng.*, 2014, pp. 895–906.
[44] M. Hausenblas and N. Bijnens. (2015). *Lambda Architecture*. [Online]. Available: http://lambda-architecture.net/.Luettu
[45] M. Httermann, *DevOps for Developers*. New York, NY, USA: Apress, 2012.
[46] H. van Vliet and A. Tang, "Decision making in software architecture," *J. Syst. Softw.*, vol. 117, pp. 638–644, Jul. 2016.
[47] L. Bass, *Software Architecture in Practice*. London, U.K.: Pearson Education, 2007.
[48] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Hoboken, NJ, USA: Wiley, 2011.



JITONG ZHAO was born in Liaoyang, China, in 1992. She received the B.S. degree in software engineering from Tongji University, Shanghai, China, in 2014, where she is currently pursuing the Ph.D. degree in software engineering. Since 2014, she has been a Research Assistant with the School of Software Engineering, Tongji University. Her research interests include software architecture and evolution, open source repository mining, data mining, machine learning, and graph mining.



YAN LIU received the Ph.D. degree in computer science from Fudan University, China, in 2005. She is currently an Associate Professor with the School of Software Engineering, Tongji University, China. From 2009 to 2010, she was a Post-Doctoral Researcher with the Fraunhofer Institute for Experimental Software Engineering, Germany. Her research interests include complex system engineering, data-intensive system architecting, domain-driven data analytics, and software repository mining. She has authored over 50 journal articles and conference papers in these areas. She served as a Program Committee Member and the Session Chair for several international conferences.

PENG ZHOU received the B.S. degree in computer science from Nanjing Tech University, Nanjing, China, in 2013, and the M.S. degree in software engineering from Tongji University, Shanghai, China, in 2017. He is currently a Senior Data Analyst with Discover Financial Technology Company, Shanghai. From 2017 to 2018, he was an Associate Researcher with the Advertising Engine and Data Algorithm Center, Online Media Group, Tencent Co., Ltd. His research interests include topic modeling on APIs of source code repositories, criminal network analysis with interactive strategies, and model-based reinforcement learning leveraging on structured semi-parametric models.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

AQ:1 = Please provide the department name for the current affiliation of all the authors.

AQ:2 = Please confirm whether the postal code for Tongji University is correct as set.

AQ:3 = Author: Please confirm or add details for any funding or financial support for the research of this article.

AQ:4 = If you haven't done so already, please make sure you have submitted a graphical abstract for your paper. A Graphical Abstract is a figure that gives an overall summary of your paper (in addition to the abstract). Please choose a figure from the paper and supply a caption at your earliest convenience for the graphical abstract. Note that captions cannot exceed 1800 characters (including spaces). If you submitted a video as your graphical abstract, please make sure there is an overlay image and caption. Overlay images are usually a screenshot of your video that best represents the video. This is for readers who may not have access to video-viewing software. Please see an example in the link below: http://ieeeaccess.ieee.org/submitting-an-article/

AQ:5 = Please note that there were discrepancies between the accepted pdf [Framing a Sustainable Architecture for Data Analytics System An Exploratory Study.pdf] and the [Framing a Sustainable Architecture for Data Analytics System An Exploratory Study.tex] in the sentences on lines 28, 29, 50–52, 73–76, 142–152, 249–255, 258–272, and 860–863. We have followed [Framing a Sustainable Architecture for Data Analytics System An Exploratory Study.tex].

AQ:6 = Please provide the publisher name and publisher location for ref. [16].

AQ:7 = Please confirm the author name and title, and also provide the organization name, organization location, and report no. for ref. [22].

AQ:8 = Current affiliation in biography of the author ''Peng Zhou'' does not match first footnote. Please check.