Big Data Analytics Has Little to Do with Analytics

Fethi Rabhi[1], Madhushi Bandara[1], Anahita Namvar[1], Onur Demirors[1,2]

[1] School of Computer Science and Engineering,
The University of New South Wales
Sydney 2052, Australia
[2] Department of Computer Engineering,
Izmir Institute of Technology
Izmir, Turkey
{f.rabhi; k.bandara; o.demirors}@unsw.edu.au, anahita.namvar@gmail.com

Abstract. As big data analytics is adapted across multitude of domains and applications there is a need for new platforms and architectures that support analytic solution engineering as a lean and iterative process. In this paper we discuss how different software development processes can be adapted to data analytic process engineering, incorporating service oriented architecture, scientific workflows, model driven engineering and semantic technology. Based on the experience obtained through ADAGE framework [1] and the findings of the survey on how semantic modeling is used for data analytic solution engineering [6], we propose two research directions - big data analytic development lifecycle and data analytic knowledge management for lean and flexible data analytic platforms.

Keywords: Data Analytic Process, Solution Engineering, Knowledge Modelling, Analytic Life Cycle

1.Introduction

Big data analytics can be defined as the process of extracting meaning from big data using specialized software systems. As the definition emphasises, it has three significant aspects: the nature of the data, the software utilized and the processes applied. The nature of big data refers to voluminous datasets often in the range of terabytes and petabytes whose size and characteristics extend beyond the ability of standard storage and computing capacity. Big data has distinct characteristics with respect to the *Volume:* the rate at which data is generated, *Velocity*: the rate at which data flows from different sources and the rate at which the produced data can be processed at maximizing its value, and *Variety*: the diversity in data types and their representation. Some challenges associated with big data can be listed as handling the massive amount of information streams generated from different sources, identifying information that is critical for decision-making, handling volatile business context and frequent changes in data and the ability to anticipate and respond on different trends.

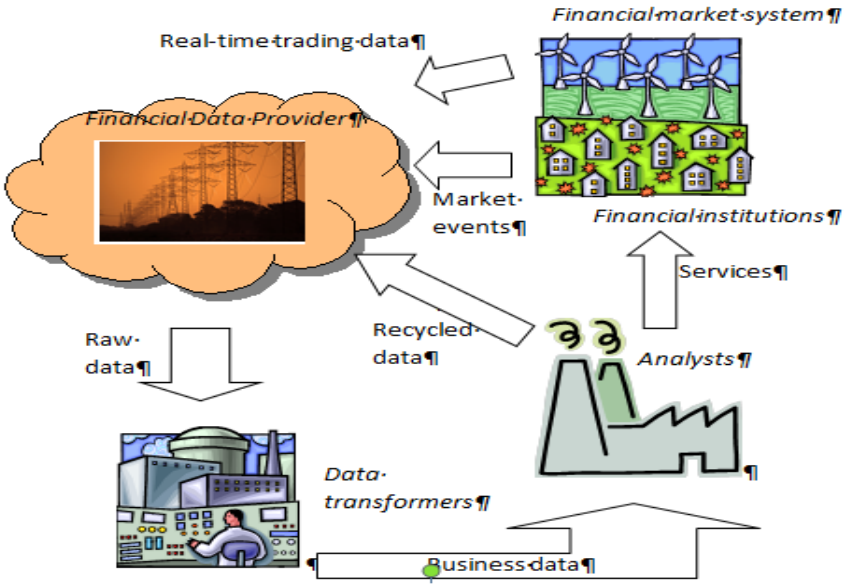

Fig 1: Financial Data Eco System

In the context of this paper, we define the big data related environment as a combination of three systems: data source, a data publisher and value generator. It differs from the traditional data warehouse environment that always has a shared view of data. Big data environments can have multiple data sources such as Internet of Things (IoT) systems, different software applications, and social networks. These sources generate data which is stored and disseminated through different data providers. Analysts can use the data published by data providers to conduct analysis and generate value out of them. The results can be used for a variety of purposes. If we take an example of the financial data eco system as shown in Figure 1, financial institutions (e.g.- banks) and financial market systems (e.g.- Australian Stock Exchange) generate different data sets which are collected, stored and published by financial data providers such as Thomson Reuters. A data scientist can access the raw data, transform them and conduct analysis to derive insights on data useful for financial institutions in their decision making. The outcome of the analysis can also be published and shared again as a new data set through the data provider.

Data analytics requires a complex process and involves multiple steps such as business understanding, data acquisition, cleaning and pre-processing, integration, pattern recognition, analyzing and interpreting results. As with the production of any service or artifact, cost, timeliness and quality determines the success of the analytics solution. Although it is depicted as an engineering solution, the analytics processes and the utilization of tools are frequently conducted in an ad-hoc fashion, based on the experience of individuals and have no traceability. Such an approach could have been feasible for the analytics problems of the last decade, but today the demand and criticality of the requirements have already gone far beyond what can be achieved with ad-hoc analytics models.

In this paper we provide our observations on how systematic approaches can improve the success rates of data analytics projects. In Section 2, we outline the role of the field of software engineering based on lessons learnt during the last 5 decades. Section 3 provides an overview of new and emerging tools, techniques and systematic approaches for handling unstructured problems as is the case for big data analytics. In the conclusion, we have summarized our observations in two aspects: analytic solution development lifecycle and better knowledge representation.

2. Why Software Engineering Matters

2.1 The Knowledge Silos Problem

To build a big data analytic solution, it is necessary for experts coming from different domains to be able to work together. One data analytic application may require application expert, social science expert, domain expert, big data specialist, statistical analytic and data mining specialists as well as a software engineer familiar with different platforms and programming techniques (see Figure 2). On one hand there are domain experts who understand the context, purpose and business value of the analytics solution. On the other hand, analytic experts specializing in statistical modelling, machine learning and mathematics are needed. Deploying solutions on an IT infrastructure requires software engineering knowledge such as data modelling, algorithms, modular design and abstraction which domain experts and analysts do not possess.

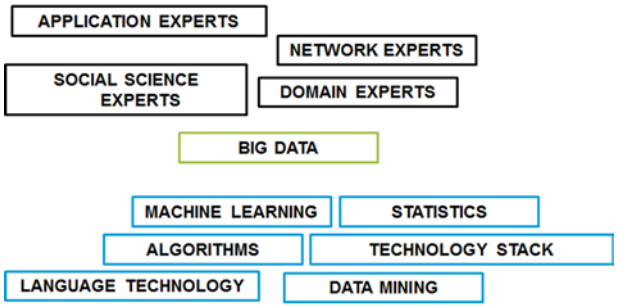

Fig 2: Big data analytics expertise silos

There is no lack of software and tools to conduct a particular data analytic task. As an evidence, observe the data analytic software stack proposed by Milosevic et al. [22] in figure 3. There are sets of platforms suitable for different levels of data analysis and tools within one layer provide same or similar services.

In many organizations, big data analytics practices are largely driven by analysts who tend to have expertise in using specific analysis or statistical modelling packages [e.g.-Weka, Tabula, SAS, Matlab]. Hence, the analysts are reluctant to design flexible analytics processes that align with organization's IT infrastructure, specific objectives, and to use a mix of data sources and software frameworks. Most

organizations rely on a manual process to integrate different analytics tasks and data elements [7,8] which are expensive and hard to maintain in the long term [7]. Moreover, according to No-Free-Lunch theorem [9], there is no one model that works best for every problem and depending on the application context and input data, analysts have to experiment with different analysis techniques to obtain optimum results.

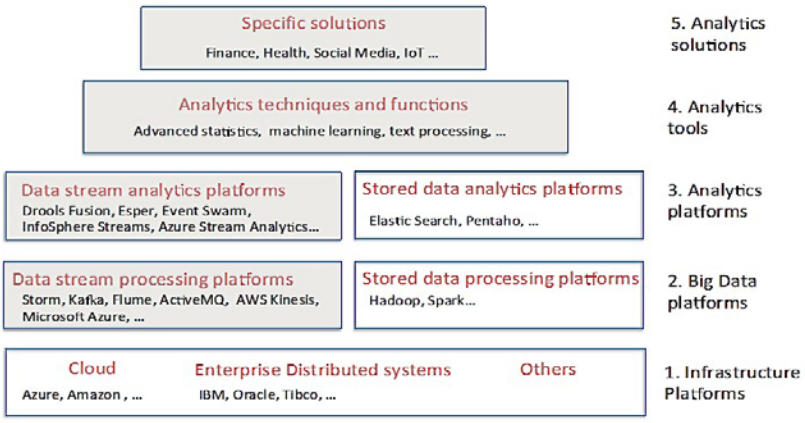

Fig 3: Analytic Software Stack [22]

Although there are many tools and techniques that are usable at different levels of the analytic solution development process, there are only few approaches that support the overall development process dynamically. Most research efforts concentrate in one area or domain such as text mining from social media or stock-market event analysis, but there is a lack of "end-to-end" methods for engineering big data analytics solutions, with proper separation of concerns.

2.2. Example of a Complex Analytics Process

To illustrate the challenges associated with data analytic processes, we exemplify a case related to predictive analytics. The process of predictive analytics aims to forecast future outcomes based on existing historical data to drive better decision [30]. In other words, it can help to identify unexpected opportunities and forecast problems before happening. In practice, predictive analytics can address business problems related to multiple disciplines from churn prediction to recommender systems. It can also anticipate when factory floor machines are likely to break down or figure out which customers are likely to default on a bank loan. Predictive analytics comprise a variety of statistical techniques and machine learning methods. Considering the inherent characteristics of predictive analytics in all domains the generic process is shown in Fig 4, However, depend on application context and input data different techniques can be applied at each stage.

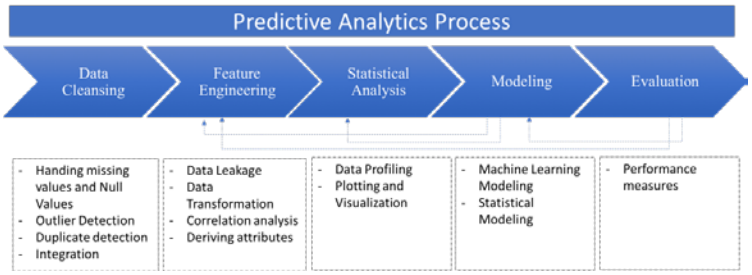

Fig 4: Predictive Analytic Process

.

Table 1: Stages of Credit Risk Prediction Process

Predictive Analytics Process		**Credit Risk Prediction**
Data Cleansing	Handling Missing Values	Removing missing values (empty, Null, N/A, none)
	Outlier Detection	Removing outliers by applying IQR method
Feature Engineering	Data Leakage	Identification of features that are not available at the time of reviewing the applicant's request for a loan, and removing them from our analysis.
	Data Transformation	Encoding ordinal features to numeric feature Binarizing nominal features Log Transformation for features with high skewness Normalization and Standardization to have measurements to a standard scale
	Correlation analysis	Applying Pearson Correlation analysis for presenting the relationship of features (predictors) with respond variable (dependent variable) which is Loan Status Investigating significant difference in predictive features between the default and non-default borrowers
	Deriving attributes	Deriving different ratios by available features. According to classification result, defined ratios increased the classifier performance
Statistical analysis	Data Profiling	Summarizing dataset through descriptive statistics such as mean, max, min, standard deviation and range
	Plotting and visualization	Depicting variables by presenting them on different plots and histograms
Modeling	Statistical Modeling	Applying Linear Discriminant Analysis and Logistic Regression analysis for predicting borrower's status
	Machine learning Modeling	Developing classification models such as Decision Tree Classifier and Random Forest for identification of default and non-default borrowers
Evaluation	Performance Measures	Considering confusion matrix, performance metrics such as False positive rate, accuracy, sensitivity and specificity has been addressed, Also ROC curve and AUC has been employed.

One such application of predictive analytics is Credit Risk Prediction, where the goal is to predict true creditworthiness of potential borrower. Table 1 depicts the general process that needs to be adapted for credit risk domain.

In practice, each of these analytic stages are conducted utilizing scripts or specific tools and integrating the data and analytic tools are done through scripts. Moreover, multiple experts should come together to understand and select data, to write software to clean and analyze them, to understand statistical and analytical models suitable for the task etc. This process is complex, time consuming and may have to go through multiple iterations before the model satisfies the evaluation criteria. Then the deployment and maintenance of the model in the bank environment should be conducted as a joint effort between system engineers, domain experts and analysts.

3. New and Emerging Software Engineering Approaches for Big Data Analytics

We discuss in this section some existing research areas in the software engineering space and their relevance in the field of big data analytics from different perspectives.

3.1 Development Processes

The best starting point for looking at the big data analytics processes from the lenses of a lean business is as an evolution of the software development life cycle models. Adapting an approach similar to Agile development can improve the analytic process by bringing a mixture of IT and business roles, providing rapid time to market strategy to model and evaluate analytic models, accepting failures and improving upon them and by challenging the existing practices. More specifically, the engineering of a big data analytics solution following an Agile method allows extensive collaboration, flexibility, and rapid development that fit with lean business practices.

We can identify three software engineering practices suitable for data analysis processes: business requirement analysis, solution design and implementation. Business requirement analysis focuses on capturing domain knowledge and acquiring requirements from different stakeholders and defining functional and non-functional requirements. Design enables the design of artifacts to be produced/discussed at a high level, with no commitment to any technology or platform. The implementation allows testing and refining the analytic solution and validating the quality. Figure 5 illustrates how different analytic expertise we discussed in section 2 are involved in these three stages of a typical Agile iteration flow.

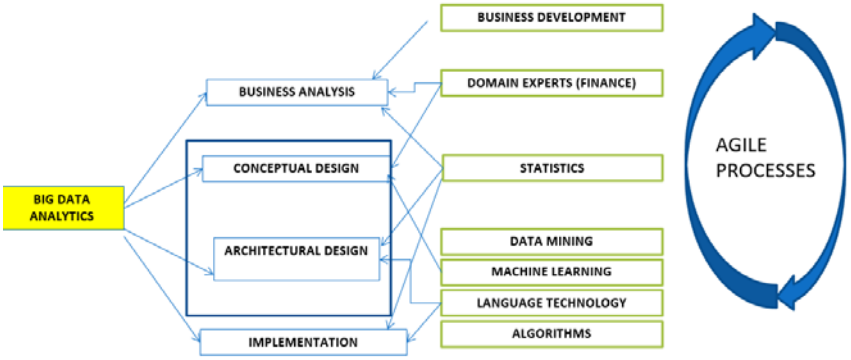

Fig 5: How expert knowledge can be leveraged in different stages of agile big data analytics process

Agile methods are particularly suitable for big data analytics problems. As the problems cannot be formulated before the solution emerges, the early feedback loop between users and engineers are critical. The iterative nature of agile methods enables to establish a systematic engineering approach while at the same time keeping the bottom up feedback loop in place.

Literature such as such as CRISP-DM [2] and Domain-oriented data mining [3] is advocating the importance of considering practices related to analytics and establishing good understanding of data to build better analytic solutions more effectively. Significant limitations observed in data analytic solution engineering space are a lack of high-level architectural and data models to understand how to compose analytic pipelines, how data should flow between the different stages and how to create mappings between the stages and appropriate tools and data sets in the underlying infrastructure.

3.2 Architectural Design

Effectively designing, building and maintaining flexible data analytics processes from an architectural perspective remains to be a challenge. Service oriented architecture and scientific workflow techniques address the issue to a certain extent by providing modular, pluggable software components and a composition environment for them. Workflow technology as applied to big data analytics is generally called scientific workflow technology. It can assist in the composition of hundreds of distributed software components and data sources. Scientific workflow technology can be used to model scientists' analysis processes, where each step typically corresponds to an individual activity or task. If each task is performed by a component (or a service), then the composition of a set of components would be equivalent to performing a sequence of tasks, where the sequence is determined by the scientific workflow model. A scientific workflow system enables the definition, management and execution of scientific workflow models and allows scientists to automate the

execution and management of complex sets of computations and data analyses, thereby enabling science at a large-scale.

Service-oriented architecture (SOA) is an architectural approach that advocates the creation of software components as autonomous, platform-independent, loosely coupled services that can be easily combined within and across enterprises to create new software applications to meet a business or scientific need [31]. Service-oriented technologies have a well-defined set of interfaces and consistent access protocols we can use to engineer data analytic solutions. In addition, business processes technologies can be used to provide an end-to-end analytic solution for the users by enabling automated or semi-automated service selection and composition. The concept of "data and analytics as a service" stems from a design paradigm of which design principles are governed by Service Oriented Architecture (SOA) [32] This concept advocates accessing data and tools "where they live" – the actual platform on which the resource resides should not matter. Therefore, service-oriented design can play an important role in linking the analytic solution design to its implementation. We identify two types of services we can leverage:

- Data services: hide data complexities and provide access to the data
- Analytics services: hide underlying technologies and conduct the model building and execution for the users

Although the use of SOA has improved interoperability, orchestration of web services into a workflow can be equally challenging for the end-user. Hence the literature emphasizes the necessity of better knowledge management in enterprise data analytic [2, 11] and scientific workflow [10] for better analytic platform development.

3.3 Integrated Frameworks

This section discusses integrated approaches for designing big data analytics processes. They generally fall under the category of model driven software development because they focus on models as central artifacts to provide an abstraction of a real-world application or system and apply model transformations to realise software systems from these models. Model Driven Engineering (MDE) is defined as the vision of constructing a model of a system that then can be transformed into a real artefact [24]. Use of MDE in the context of service-oriented architecture can deliver powerful software engineering methods [25].

One way to provide a platform for end-to-end data analytic solution development is to follow an MDE approach where knowledge related to data, mining algorithms and analytic services are captured through models which are leveraged to derive an analytic solution. There is ample literature emphasizing the advantage of using models [23,28,29], in analytic solution space to model data, analytic requirement or services etc. There are only a few studies in the literatures such as Rajbhoj et. al [26] and Ceri et al. [27] that explores the potential of applying MDE for big data analytics, but they are limited to particular analytic tool or technology such as Map-Reduce framework [26].

The ADAGE framework [1] specifically leverages the capabilities of service-oriented architectures and scientific workflow management systems into data analysis. The main idea is that the models used by analysts (i.e. workflow, service,

and data models) contain concise information and instructions that can be viewed as an accurate record of the analytics process, become a useful artefact for provenance tracking and ensure reproducibility of such analytics processes. As shown in Figure 6, the ADAGE framework consists a set of architectural patterns and operational guidelines.

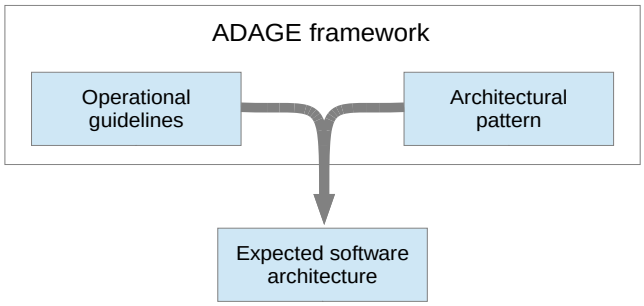

Fig 6: Adage Framework

ADAGE architecture patterns support the definition of analysis processes in a more convenient manner than using generic and conventional business processes. It uses a reference data model closely associated with a target domain to standardize the representation of datasets. Adage framework uses a set of services to process the datasets, so as to transform them into other datasets or information. Both the reference data model and the ADAGE services are embedded in a service-oriented architecture (SOA). Figure 7 represents a definition of an end-to-end data analytic process from importing data into dissemination of findings, defined using the ADAGE architectural pattern. Figure 8 represents an application of that analytic process for the analysis of financial market data.

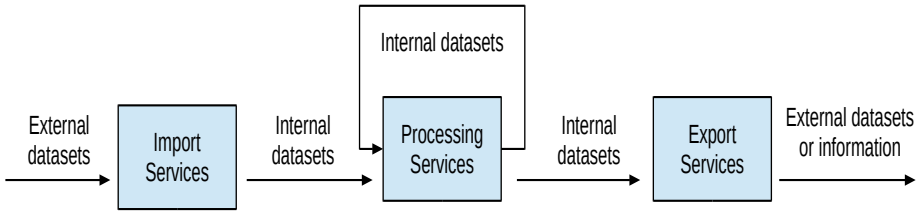

Fig 7: Definition of an analytic process through the ADAGE architectural pattern

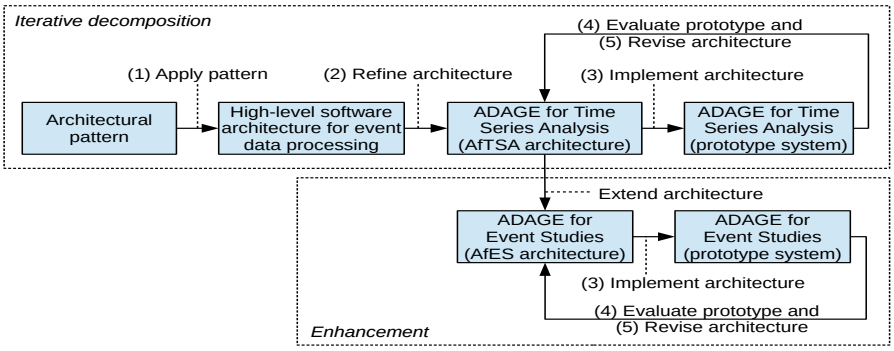

Fig 8: Application to the analysis of financial market data

However, defining suitable data models to accurately represent complex business contexts associated with an analytic problem is not easy.

3.4 Knowledge Representation

The main critique of existing MDE approaches is that they often assume simple data reference models, which is unrealistic, hard to evolve and difficult to create and maintain when there are multiple stakeholders with conflicting viewpoints. Any analytic system has to recognize that different types of mental models can co-exist, each type of model can be particular to a community of practice, the mappings between concepts from different models can be subjective and the reference model needs to allow different interpretations of the data by different people. As an example, a financial data analysis system can have two types of models: event model and time-series model, two communities: computer science and statistics and it is not possible to always map between raw data and variables consistently.

Semantic technology, which is based on the vision of semantic web by Tim Berners-Lee is a new approach for modelling knowledge, data as well as their semantics and there is a well-developed set of standards and notations: RDF, RDSFS, OWL, supported by different tools for modelling, storing, querying, and inferencing the knowledge. Different communities have adapted semantic technologies to build standard ontologies related to their practices (e.g. ResMED for medical domain and FIBO in Finance).

The work in [21] summaries the value of semantic technology and ontologies from three angles. 1. Ontology is a way of clarifying meaning and reducing unnecessary complexity (e.g.- a precise technical jargon) 2. Ontology is a way to improve agility and flexibility, 3. Ontology is a way to improve interoperability and integration by representing information consistently across multiple domains and machines.

The main role of an ontology is to capture the domain knowledge, to evaluate constraints over domain data, to prove the consistency of domain data and to guide domain engineering while developing domain models [5]. Pan et. Al. [4] discuss in-depth about how a generic software development process can be enhanced with the use of ontologies as ontologies provide a representation of knowledge and the relationship between concepts they are good at tracking various kinds of software development artefacts ranging from requirements to implementations code [4]. Such enhancements are important for the domain of data analytic where analyst have to

deal with heterogeneous data sets, analytic models, and continuously changing requirements to derive different insights from big data sets.

Though there is multiple work done leveraging the semantic technology for analytics, they do not provide a complete solution that can address the challenges faced by analysts. Early work looked at how semantic web technology helps information integration [12]. Moreover, there is a body of work that uses semantic web technologies for Exploratory OLAP [13], mainly to address the heterogeneity of data. There is a lot of work done on introducing semantics to scientific workflows such as SADI [17] and WINGS [18], to discover services that meet user requirements. Yet they do not discuss how the domain knowledge can be captured and how the whole process of the analytics can be automated and made user-driven. The existing work related to semantic web services (OWL-S, WSDL, WSMO etc.) plays a prominent role in service composition, yet they look at the operation angle and does not support the incorporation of analytic domain concepts.

Many existing applications that apply semantic technology in data analytics are very limited to a single domain and it is difficult to generalize and adapt them to design reusable architectures. For example, [14,15] are limited to urban data, [16] applies for agriculture domain. Largely these applications were designed and developed in isolation, specific to a particular need of an organization or entity. Moreover, the solutions are highly domain specific and extendibility for new use cases or adaptability of them in other domains are questionable.

Work of Barisson and Collard [19] and Kumara et.al. [20] are focusing on using semantic technology for CRISP-DM [2] based data mining process. However, they lack the linkage between the domain knowledge and analytic tasks and proposed models are complex to understand, less generalizable and difficult to be used for end-to-end analytic process development.

4. Conclusions and Future Work

From the discussions in the previous section, we believe that software engineering has a lot to offer in improving big data analytics solution. We identify two key areas of research work:

- new big data analytic development lifecycle
- better knowledge representation for analytics.

4.1 Towards Big Data Analytics Development Lifecycle

First we advocate the creation of a new big data analytic development process that maps different stages of analytic process into those of software engineering such as requirement analysis, design and implementation. This process is iterative and may follow multiple iterations to come up with the final solution. The activities followed under three stages are illustrated in Figure 9.

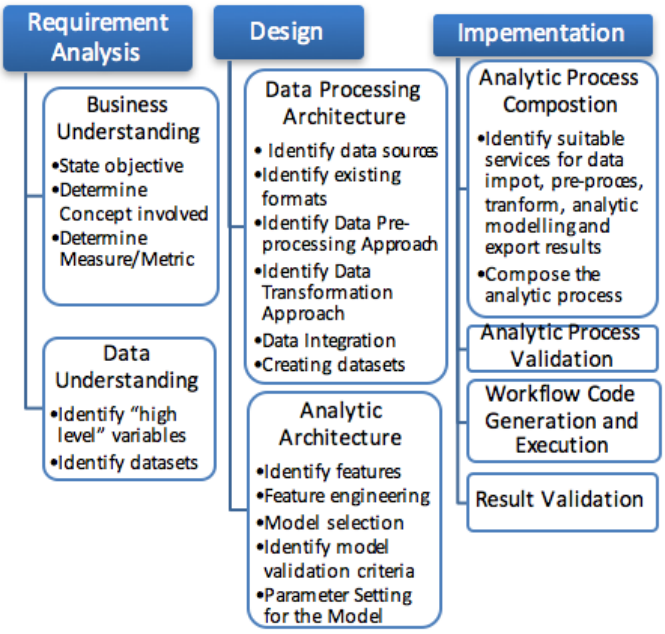

Fig 9: Big Data Analytics Development Cycle

At the requirements analysis stage, analyst go through analyzing the problem; understanding the business domain and what is the context and nature of the data available. Design stage consists of two parts: data processing architecture design and analytic architecture design. These parts are interrelated as the nature of data influences the kind of model suitable for the analytic problem and also the data pre-processing and transformation should cater the input requirements and formats of the selected model.

At the implementation stage, we suggest to leverage service-oriented architecture and workflow modeling to conduct analytic process composition and execution.

4.2 Better Data Analytics Knowledge Management

We identify the need and propose a framework where analytic process, services and scientific workflows represented by semantic technology as well as domain knowledge fits together to provide efficient event data analytic platforms. To explore the state-of-art that use semantic technology for data analytic solution engineering and identify its potential, we conducted a systematic literature review that explores literature spread over three spheres: software engineering, semantic modelling and data analytics. A detailed discussion of this review findings is presented in [6].

Through the review we answered to the questions about what knowledge related to data analytic process is captured by existing work- we identified four classes of

semantic concepts: Domain, Analytic Service and Intent. Then we study how this knowledge (semantic concepts) is applied in analytic process development process, related to different development tasks such as business understanding, data extraction, model selection and analytic process composition. Based on the limitations we found form the literature survey [6] we suggest future research directions in knowledge enabled analytics. Mainly, the analysts should consider leveraging intent related models that represent business requirements and goals, as only then the solution can address the core problem. Furthermore, model building should not be an isolated task of trial and error. Analysts can leverage different analytic models to understand available model building methods and instantiate them. Semantic models are useful in each stage of the analytic process, but state-of-art is limited to use them for a specific task such as data integration or model selection. Hence it is necessary to have good models that contain sufficient knowledge to help analysts throughout the development process.

The survey [6] provides evidence to the importance of service based approaches in analytic solution engineering and the SOA community has multitude of research regarding the service modelling, selection etc. which are useful for realizing the Agile based big data analytics development cycle. Furthermore, the work emphasizes the significance of model driven analytic solution engineering, which we try to cater through the big data analytics development lifecycle by introducing implementation as the third stage and facilitating process composition. Process composition and execution can be of model-driven fashion once the good models are in place, for incorporating SOA and workflow technologies. Data quality governance is a main concern that needs to be addressed when realizing model driven and service based analytic platforms. This can be the starting point for providing analytics as a service where expert knowledge is captured and provided for anyone to compose their own analytic solution.

Data analytics, domain expertise and software engineering communities need to work together to design ontologies that can support end-to-end data analytic solutions. Involvement of all three expert groups will result in better ontologies and it will aid to preserve the analytic related knowledge which exists in isolation today.

Finally, we emphasize the necessity of incorporating analytics as part of value chain of a business, rather than treating it as an isolated tool used by scientists. To realize this objective, analytic technologies should align well with the infrastructure of the organization and flexible to cater changing business values. We believe that the Agile lifecycle and the knowledge management strategies that we advocate can provide means to realize effective integration of business, IT and analytic environments within an organization.

References

1. L. Yao and F. A. Rabhi, “Building architectures for data-intensive science using the adage framework,” Concurrency and Computation: Practice and Experience, vol. 27, no. 5, pp. 1188–1206, 2015.
2. P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, “Crisp-dm 1.0 step-by-step data mining guide,” 2000.

3. G. Wang and Y. Wang, “3dm: domain-oriented data-driven data mining,” Fundamenta Informaticae, vol. 90, no. 4, pp. 395–426, 2009.
4. Pan, Jeff Z., Steffen Staab, Uwe Aßmann, Jürgen Ebert, and Yuting Zhao, eds. *Ontology-driven software development*. Springer Science & Business Media, 2012.
5. F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider (eds.), The Description Logic Handbook: Theory, Implementation, and Applications (Cambridge University Press, Cambridge, 2003). ISBN 0-521-78176-0
6. Madhushi Bandara, Fethi Rabhi, Semantic Modelling for Engineering Data Analytic Solutions: A Systematic Survey (In Review)
7. Espinosa, R., Garca-Saiz, D., Zorrilla, M., Zubco, J. J., Mazn, J. N. : Enabling non-expert users to apply data mining for bridging the big data divide. InInternational Symposium on Data-Driven Process Discovery and Analysis(pp. 65-86). Springer Berlin Heidelberg, (2013).
8. Fisher, D., DeLine, R., Czerwinski, M., Drucker, S.: Interactions with big data analytics.interactions,19(3), 50-59,(2012)
9. Magdon-Ismail M., No free lunch for noise prediction. Neural computation ,12(3):547-564, (2000)
10. Taylor, J. : Framing Requirements for Predictive Analytic Projects with Decision Modeling,(2015)
11. Shumilov, S.,Leng, Y., El-Gayyar, M., Cremers A. B. , Distributed Scientic Work-ow Management for Data-Intensive Applications, pp. 65-73, (2008)
12. Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hbner, S.: Ontology-based integration of information-a survey of existing approaches. In IJCAI-01 workshop: ontologies and information sharing, Vol. 2001, pp. 108-117,(2001)
13. Abell, A., Romero, O., Pedersen, T. B., Berlanga, R., Nebot, V., Aramburu, M. J., Simitsis, A.: Using semantic web technologies for exploratory OLAP: a survey.IEEE transactions on knowledge and data engineering,27(2), 571-588,(2015)
14. Puiu, D., Barnaghi, P., Tonjes, R., Kumper, D., Ali, M. I., Mileo, A.et al.,: City- Pulse: Large Scale Data Analytics Framework for Smart Cities: IEEE Access, vol. 4, pp. 1086-1108, (2016)
15. Gao F., Ali,M. I., Mileo,A.,: Semantic discovery and integration of urban data streams: Proceedings of the Fifth International Conference on Semantics for Smarter Cities, vol. 1280, pp. 15-30 (2014)
16. Laliwala, Z., Sorathia, V., Chaudhary, S.: Semantic and rule based event-driven services-oriented agricultural recommendation system. 26th IEEE International Conference onDistributed Computing Systems Workshops, pp. 24-24, IEEE,(2006)
17. Withers, D., Kawas, E., McCarthy, L., Vandervalk, B., Wilkinson,M.: Semantically- guided workow construction in Taverna: the SADI and Biomoby plug-ins: In International Symposium On Leveraging Applications of Formal Methods, Verication and Validation, pages 301-312. Springer, (2010)
18. Gil Y., Ratnakar V., Deelman E., Mehta G., Kim J. : Wings for Pegasus:Creating large-scale scientic applications using semantic representations of computational workows. In Proceedings of the 19th National Conference on Innovative Applications of Articial Intelligence - Volume 2, IAAI'07, pages 1767-1774. AAAI Press (2007)
19. Brisson, L., Collard, M.:An ontology driven data mining process. In International Conference on Enterprise Information Systems,pp. 54-61,(2008)
20. Kumara, B. T., Paik, I., Zhang, J., Siriweera, T. H. A. S., Koswatte, K. R. :Ontology-BasedWorkow Generation for Intelligent Big Data Analytics. 2015 IEEE International Conference on Web Services (ICWS), pp. 495-502,IEEE. (2015)
21. M. Uschold. Making the case for ontology. Applied Ontology , 6(4):377{385,2011.
22. Milosevic, Z., Chen, W., Berry, A. & Rabhi, F. A. 2016. Real-Time Analytics.
23. J. Taylor. Framing analytic requirements. 2017.
24. S. J. Mellor, T. Clark, and T. Futagami. Model-driven development: guest editors' introduction. IEEE software , 20(5):14-18, 2003.
25. D. Ameller, X. Burgues, O. Collell, D. Costal, X. Franch, and M. P. Papazoglou. Development of service-oriented architectures using model-driven development: A mapping study. Information and Software Technology , 62:42-66, 2015
26. A. Rajbhoj, V. Kulkarni, and N. Bellarykar, “Early experience with model-driven development of map-reduce based big data application,” in Software Engineering Conference (APSEC), 2014 21st Asia-Pacific, vol. 1. IEEE, 2014, pp. 94–97.

27. S. Ceri, E. Della Valle, D. Pedreschi, and R. Trasarti, “Mega-modeling for big data analytics,” Conceptual Modeling, pp. 1–15, 2012.
28. S. Luj´an-Mora, J. Trujillo, and I.-Y. Song, “A uml profile for multidimensional modeling in data warehouses,” Data & Knowledge Engineering, vol. 59, no. 3, pp. 725–769, 2006.
29. H. Maci`a, V. Valero, G. D´ıaz, J. Boubeta-Puig, and G. Ortiz, “Complex event processing modeling by prioritized colored petrinets,” IEEE Access, vol. 4, pp. 7425–7439, 2016.
30. A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," International Journal of Information Management, vol. 35, no. 2, pp. 137-144, 2015
31. Papazoglou, M. P., Traverso, P., Dustdar, S. & Leymann, F. 2007. Service-oriented computing: state of the art and research challenges. Computer, 38-45.
32. Thomas, E. 2007. SOA principles of Service Design. Boston: Prentice Hall, 37, 71-75.