Feature Driven Survey of Big Data Systems

Cigdem Avci Salma[1,2], Bedir Tekinerdogan[1] and Ioannis N. Athanasiadis[1]
[1]*Information Technology, Wageningen University, Wageningen, The Netherlands*
[2]*ESTEC, European Space Agency, Noordwijk, The Netherlands*

Keywords: Feature Driven Design, Feature Modeling, Big Data.

Abstract: Big Data has become a very important driver for innovation and growth for various industries such as health, administration, agriculture, defence, and education. Storing and analysing large amounts of data are becoming increasingly common in many of these application areas. In general, different application domains might require different type of big data systems. Although, lot has been written on big data it is not easy to identify the required features for developing big data systems that meets the application requirements and the stakeholder concerns. In this paper we provide a survey of big data systems based on feature modelling which is a technique that is utilized for defining the common and variable features of a domain. The feature model has been derived following an extensive literature study on big data systems. We present the feature model and discuss the features to support the understanding of big data systems.

1 INTRODUCTION

The term *Big Data* usually refers to data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time. In general, Big Data can be explained according to three V's: Volume (amount of data), Velocity (speed of data), and Variety (range of data types and sources) (Laney, 2001). The realization of Big Data relies on disruptive technologies such as Cloud Computing, Internet of Things and Data Analytics. An increasing number of industries such as health, administration, agriculture, defence, and education, adopt Big Data to support innovation and growth. Usually, these systems represent major, long-term investments requiring considerable financial commitments and massive scale software and system deployments. Hence, it is important to identify the required features for developing the proper big data system. In fact, the characteristics of big data systems may vary in terms of the application domain and as such require different Big Data properties.

In this paper, we provide a survey of big data systems based on feature modelling (Kang et al., 1990). A feature model is a model that defines the common and variant features of a domain. The proposed feature model has been derived based on the corresponding literature study of big data systems. The feature model can be of benefit for practitioners who aim to develop big data systems. Researchers on the other hand can use the feature model to scope the domain of big data systems and the related research activities.

The remainder of the paper is organized as follows. In Section 2, we present the background on feature modelling. Section 3 elaborates the feature model for big data systems. Section 4 provides the discussion and finally section 5 concludes the paper.

2 FEATURE MODELING

For understanding the domain of big data systems we have applied a systematic domain analysis process. Domain analysis process (Arrango, 1994) aims to identify, capture and organize the domain knowledge for a problem domain in order to facilitate reusability, especially during the development of a new system. (Tekinerdogan and Aksit, 2001; Kang et al., 1990). The notion of domain can be defined differently by different authors. We adopt the definition of the term "domain" in the UML glossary which is as follows: "*Domain is an area of knowledge or activity characterised by a set of concepts and terminology understood by practitioners in that area*."

Several domain analysis methods have been

Salma, C., Tekinerdogan, B. and Athanasiadis, I.
Feature Driven Survey of Big Data Systems.
DOI: 10.5220/0005877503480355
In *Proceedings of the International Conference on Internet of Things and Big Data (IoTBD 2016)*, pages 348-355
ISBN: 978-989-758-183-0

identified in the literature. Although these differ in detail they also contain identical activities (Tekinerdogan and Aksit, 2001; Czarnecki et al., 2006; Kang et al., 1990). In general we can distinguish between domain scoping (Harsu, 2002) and domain modelling (Lee et al., 2002).

Domain scoping aims to determine the domains of interest, the stakeholders, and their goals (Tekinerdogan et al., 2005). Domain modelling aims to provide an explicit representation of the domain. A domain model is derived using a commonality and variability analysis approach. Using a commonality analysis the common concepts of the identified set of knowledge sources are identified. Variability analysis focuses on identifying and modelling the differences. The domain model usually represents both the commonality and variability and can be in different forms (object-oriented language, algebraic specifications, rules, conceptual models) (Tekinerdogan and Öztürk, 2013).

Among the domain modelling approaches, feature modelling is extensively used (Araùjo et al., 2005) which is in particular used for representing commonality and variability within one diagram. A feature diagram is constructed as a tree whereby the root element represents a concept or system and its descendent nodes define the properties, the features, of the concept. Features can have sub-features and can be mandatory, optional, or alternative to each other. A feature diagram by itself thus represents the possible configuration space within a given system. In general not all configurations are possible due to some selection constraints. Hereby, two basic constrains can be identified. A *mutex* constraint defines the mutual exclusion between two concepts. That is, if a feature f1 has a mutex relation with another feature f2 then these two features cannot be selected together. On the other hand the *requires* relation between features defines that the corresponding features need to be selected together. FODA (Kang et al., 1990) describes the basic variability model and different extensions have been proposed in various studies. The classification of these approaches can be found in (Sinnema and Deelstra, 2007). In general, feature modelling has been broadly applied in software product line engineering, domain-driven design and software architecture design. In this paper we will adopt feature modelling to represent the concepts of Big Data systems.

3 FEATURE MODELING OF BIG DATA SYSTEMS

For analysing big data systems we will first consider the overall reference architecture of big data systems. A reference architecture based on (Marz and Warren, 2015) is shown in Figure 1. Similarly, several different reference architectures can be derived from the literature. The goal of this study is to describe the feature model for Big Data systems that will support the design of the big data architecture for the particular application domains.

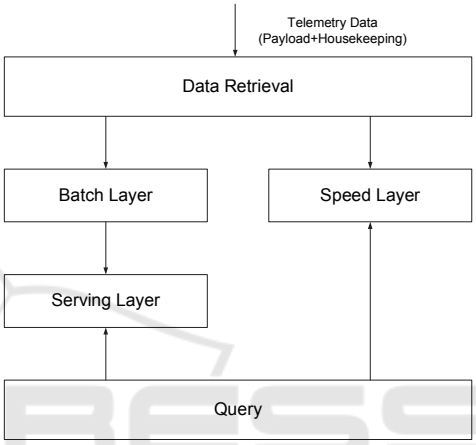

Figure 1: Conceptual Architecture of a Big Data System by (Marz and Warren, 2015).

Figure 2 shows the approach for defining a feature model for Big Data systems. The approach consists of two basic activities of domain scoping and feature modeling. In the domain scoping part we have defined the search strategy, collected the papers and selected the final set of papers that would be used for feature modelling according to our inclusion/exclusion criteria. Our search method was a manual search of specific conference or journal papers since 2010. In particular we selected papers that explicitly discussed reference architectures. The selected papers are shown in Table 1. In order to assure the sufficient coverage, papers are searched via and selected from three different electronic libraries which are IEEE Xplore, Google Scholar and ScienceDirect. Our inclusion criteria for a paper is the maximum known feature coverage and the reference architecture in each paper among the outcome paper set is analysed in terms of the feature set that it covers, in order to ensure that the maximum known feature coverage is reached without repetition.

For feature modelling we selected each paper of

Table 1 and extracted the features from the big data reference architectures that is presented in the paper. The features that have a very high frequency among the reference architectures are selected as the mandatory features where the others are considered as optional. Afterwards, the features and their relations are used to build up the feature model. The feature model was evaluated against each paper by the authors.

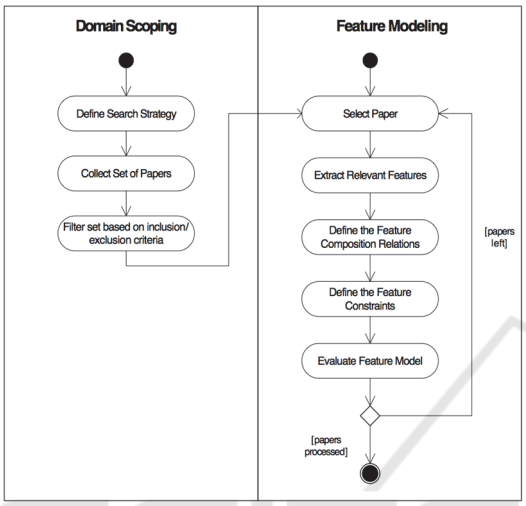

Figure 2: Approach for deriving a feature model of Big Data systems.

Table 1: List of papers to derive the features of Big Data systems in alphabetical order.

1	B. Geerdink. "A Reference Architecture for Big Data Solutions."
2	C. Ballard, et al. Information Governance Principles and Practices for a Big Data Landscape. IBM Redbooks, 2014.
3	D. Chapelle. "Big Data & Analytics Reference Architecture." An Oracle White Paper (2013)
4	M. Maier. A. Serebrenik, and I. T. P. Vanderfeesten. "Towards a Big Data Reference Architecture." (2013).
5	NIST Big Data PWG, Draft NIST Big Data Interoperability Framework: Volume 6, Reference Architecture (2014).
6	N. Marz, and J. Warren. "Big Data: Principles and best practices of scalable realtime data systems. "Manning Publications Co. (2015).
7	Oracle, Information Management and Big Data A Reference Architecture, An Oracle White Paper, February (2013).
8	P. Pääkkönen, and D. Pakkala. "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems." Big Data Research (2015).
9	S. Soares,. "Big Data Governance." Information Asset, LLC (2012).

3.1 Top Level Feature Model

The top level feature diagram of Big Data Systems that we have derived is shown in Figure 3. The legend part represents the different types of features including optional and mandatory (Aksit et al., 1999).

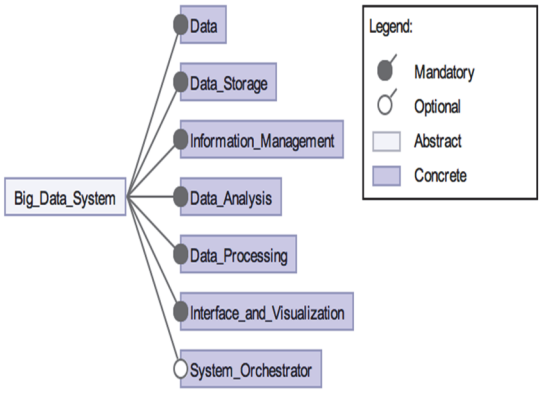

Figure 3: Top Level Feature Model.

A Big Data system consists of the mandatory features Data, Data Storage, Information Management, Data Analysis, Data Processing, Interface and Visualisation and the optional feature System Orchestrator. Among the mandatory features, although the necessity of the Information Management in Big Data Systems is evident, it has a vague representation in (Pääkkönen and Pakkala, 2015; Geerdink, 2013) only with its sub-features. In (Geerdink, 2013; Maier et al., 2013) the System Orchestrator is present with the form of a controller module in the architecture. The sub-features of the System Orchestrator are not explained in detail.

3.1.1 Data

Data is the feature that defines the variety of the data types in terms of their usage, state and representation. Hence, data in Big Data systems can be classified in five dimensions: mobility, structure, processing, security (Ballard et al., 2014) and modality. The corresponding feature diagram is shown in Figure 4. Mobility addresses the status of the data during processing and analysis activities and can be either *batch* or *streaming*. While the design of the batch processing modules should be aligned with the quality goals in terms of scalability, the design of the stream processing activities are effecting the performance of the system. In (Marz and Warren, 2015; Chapelle, 2013) mobility is referred as a feature of the reference architecture.

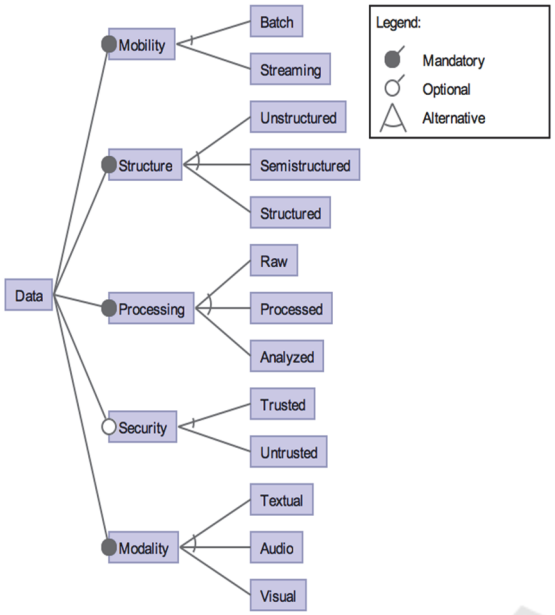

Figure 4: Feature diagram of the Data feature and its sub-features.

Another sub-feature of the feature Data in Big Data systems is the Structure. Depending on the source of the data, it can be in either of the following three structural phases: Unstructured, Semi-structured or Structured. The formation of the data processing, analysis and storage modules is highly depended on the structure of the data.

The *Processing* feature defines the processing state of the data. Initially, the data in Big Data systems is *raw*, and can be *processed* and *analysed.* From the security perspective, the data can be either trusted or untrusted (Ballard et al., 2014). Some of the possible modalities of the Big Data are textual, audio and video.

3.1.2 Information Management

This feature represents the governance of the data in terms of security, privacy, data integration and data quality. The feature is composed of three sub-features, which is typically implemented as layers which are the staging layer, access and performance layer and the foundation layer. The staging layer is an abstraction of the data acquisition process. It calibrates the data receiving rate and prepares the data for further processing. The access and performance layer is utilised for access and navigation of the data. Finally, the foundation layer isolates the data in storage from the business processes so that the data is ensured to be resilient to changes.

Especially in (Soares, 2012; Ballard et al., 2014; Chapelle, 2013; May, 2014), information management has a crucial role in the reference architecture. The well-structured information management module architecture in (Ballard et al., 2014) is used as a basis for the design of this feature. On the other hand, (Pääkkönen and Pakkala, 2015; Geerdink, 2013; Maier et al., 2013) distributes the information management features among the other high level features.

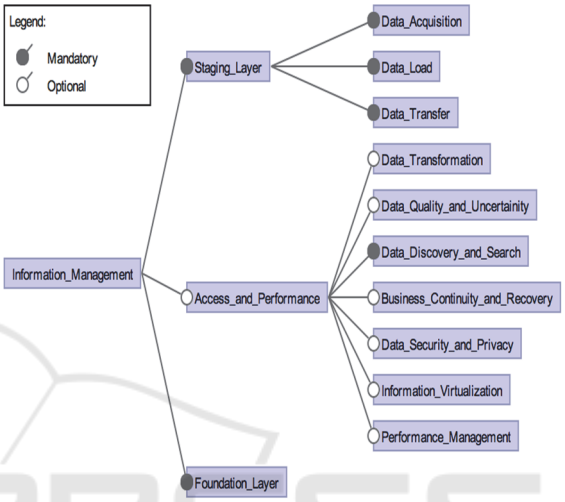

Figure 5: Feature diagram of Information Management feature and its sub-features.

3.1.3 Data Processing

Preprocessing steps such as compression aim to prepare the data so that they facilitate processing activities. Depending on the state of the data, the processing can be classified either as stream processing (f.e. filtering, annotation) or batch processing (f.e. cleaning, combining and replication). For further processing purposes, depending on the requirements of the system, information extraction, data integration, in memory processing and data ingestion activities can be employed.

As shown in Figure 6, classification, entity recognition, relationship extraction and structure extraction can be listed among the information extraction features. Data fusion, entity recognition and schema integration are the basic data integration activities. Under the category of in memory data processing as opposed to processing in the hard disks, hard speed query processing and results caching are the possible features to be utilised. Furthermore, data ingestion which involves data obtaining and processing activities for later use is

another optional data processing feature.

In (Chapelle, 2013), the data processing concept is discussed considering the query processing concerns and in memory processing is emphasised. Apart from (Maier et al., 2013; Chapelle, 2013), data integration concepts are not addressed.

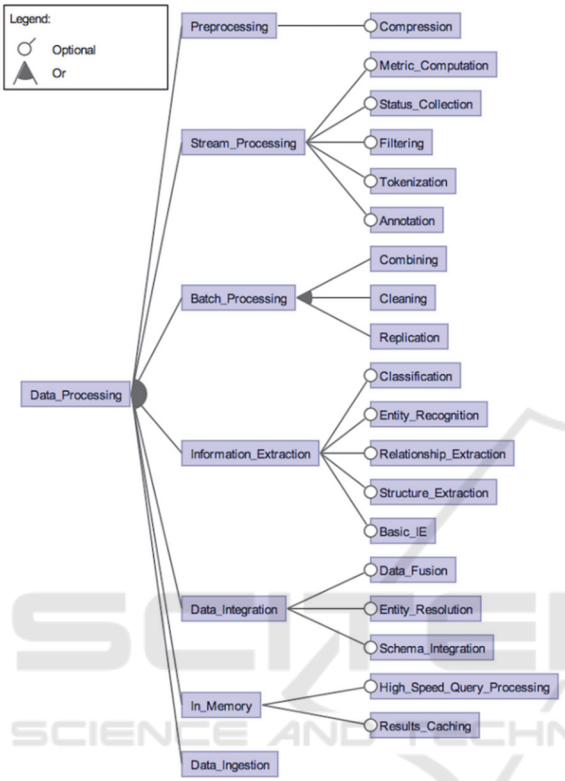

Figure 6: Feature diagram of Data Processing feature and its sub-features.

3.1.4 Data Analysis

Data analysis is one of the major features of the Big Data system. Stream analysis, high level analysis, ad hoc analysis, event processing and deep analysis are its sub-features that take part in the reference architectures in the known literature.

Stream analysis makes use of methods such as online computation, nearline computation and ranking algorithm to analyse the data streams. Statistical analysis, text mining, data mining, geospatial analysis and time series analysis are considered as high level analysis strategies in (Ballard et al., 2014). Another sub-feature, which is ad hoc analysis, is presented in (Maier et al., 2013) and defined as the module that presents the user the possible analysis strategies so that s/he can construct an ad-hoc data analysis process. Event detection/action (processing) is also listed among the data analysis modules in (Geerdink, 2013) (Ballard et al., 2014; Chapelle, 2013). Finally, gathering information from multi-source datasets with structured and unstructured data, that is known as deep analysis, is presented among the data processing modules with the querying and optimization methodologies (Pääkkönen and Pakkala, 2015; Chapelle, 2013). In Figure 7, the data analysis feature and its sub-features are presented.

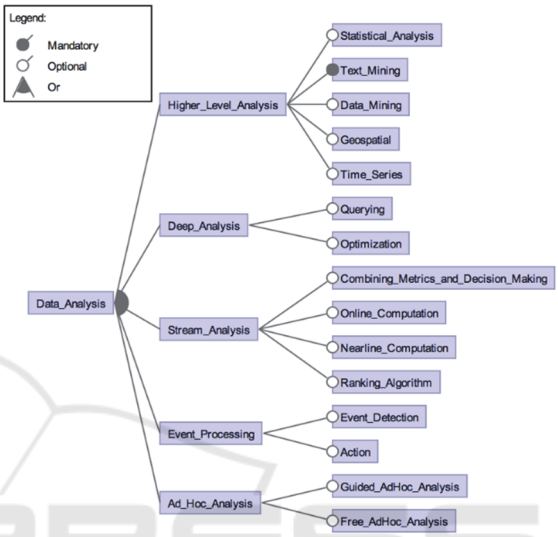

Figure 7: Feature diagram of Data Analysis feature and its sub-features.

3.1.5 Data Storage

Data storage feature consists of query processing, indexing, distributed file system, data model, metadata management and database sub-features which are shown in Figure 8. Besides the traditional data storage features such as metadata management, relational model and relational database, a Big Data system can also employ features for streaming data (i.e. in-stream query processing) and for storing various data structures, with non-relational models (i.e. NoSQL).

Although metadata management is an important asset for data storage, it appears that except (Marz and Warren, 2015; Soares, 2012) none of the reference architectures prescribe a metadata management module. In (Pääkkönen and Pakkala, 2015), the data storage modules are derived by means of the applied processing strategy. The related modules in (Maier et al., 2013) are eligible for information management feature in our feature model. Moreover in (Geerdink, 2013), the storage modules are differentiated according to the data source, timeliness or the target module that can

process the data. The data models and distributed file systems are addressed in (Marz and Warren, 2015; Ballard et al., 2014).

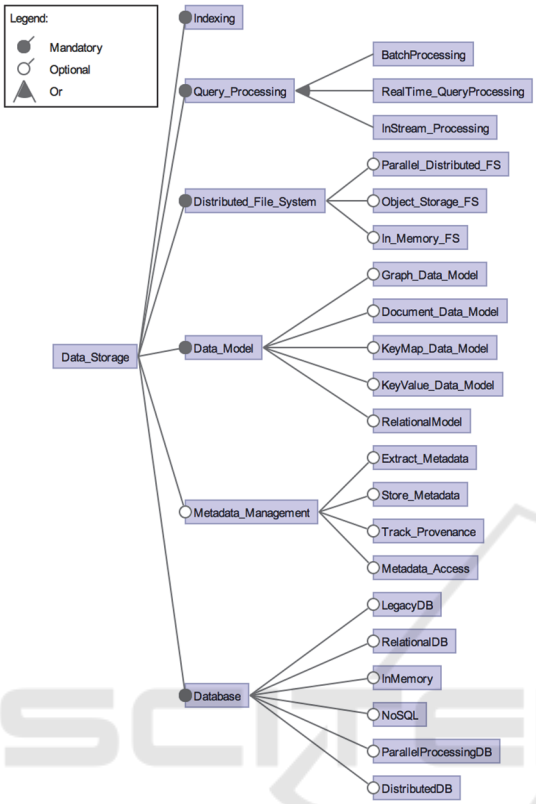

Figure 8: Feature diagram of Data Storage feature and its sub-features.

3.1.6 Interface and Visualization

This feature provides the interaction of the Big Data system with the user and other applications. The sub-features of the Interface and Visualization feature are expanded in Figure 9.

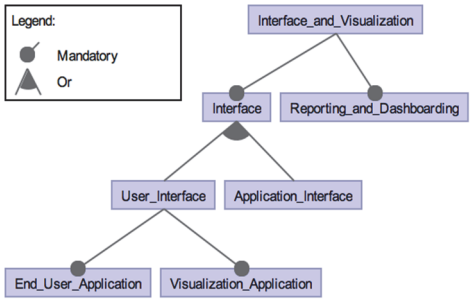

Figure 9: Feature diagram of Interface and Visualization feature and its sub-features.

While reporting and dashboarding feature is used for only information presentation, the user interface and application interface features can provide interactive services for the users and applications storage. The retrieval of the data via the user interfaces will require interactive response which needs advanced optimisation techniques (May, 2014).

3.2 Feature Characterization of Existing Systems

In the previous section we have defined the family feature model for Big Data systems. The family feature model defines the different possible Big Data systems. We can use the family feature model to characterize a particular Big Data system by selecting the corresponding features. In this section, we will illustrate this for two Big Data systems namely Facebook and Twitter.

3.2.1 Facebook

The software architecture of Facebook is discussed in (Thusoo et al., 2010). By means of the Big Data System feature model that we have presented in Section 3.1, the architecture covers all mandatory features that are Data, Data Storage, Information Management, Data Analysis, Data Processing, and Interface and Visualization, while System Orchestrator is not a feature of the Facebook software architecture. Moreover, as shown in Figure 10, Event Processing, which is a sub-feature of Data Analysis, is not included among the capabilities of the architecture. Similarly, Data Integration sub-feature of the Data Processing feature that is achieved by Data Fusion, Entity Resolution or Schema Integration is not described as a part of the architecture. Finally, via the derived feature model

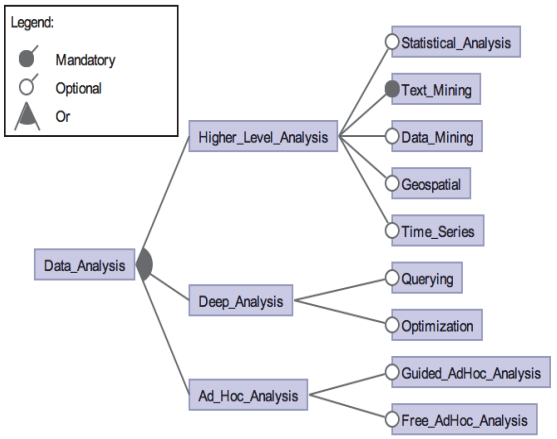

Figure 10: Feature diagram of Data Analysis feature of Facebook.

of Facebook, it is observed that the structure and the mobility of the data is addressed while the modality and security perspectives of the data in the architecture is not clarified.

3.2.2 Twitter

The Twitter data is often streaming and unstructured (Mishne et al., 2013). Data security does not have an impact on the architectural design. Similarly, the modality does not have a significant role and the architectural components are specialized not for visual or audio but textual data. Query processing is one of the main functionalities of the system where supporting optimization strategies are applied to. Especially real-time query processing is the main objective of the system while the architecture also meets the sufficient batch processing requirements. Besides, the system employs metadata management. The data integration capability of Twitter is limited. The first implementation of the Twitter system was Hadoop-based, which did not meet latency requirements. Therefore in the second implementation, a custom in-memory processing engine is employed. Ingestion is not mentioned in the architectural descriptions. The Data Processing feature of Twitter is presented in Figure 11.

In terms of data analysis, besides the querying functionalities, optimization is also emphasized in

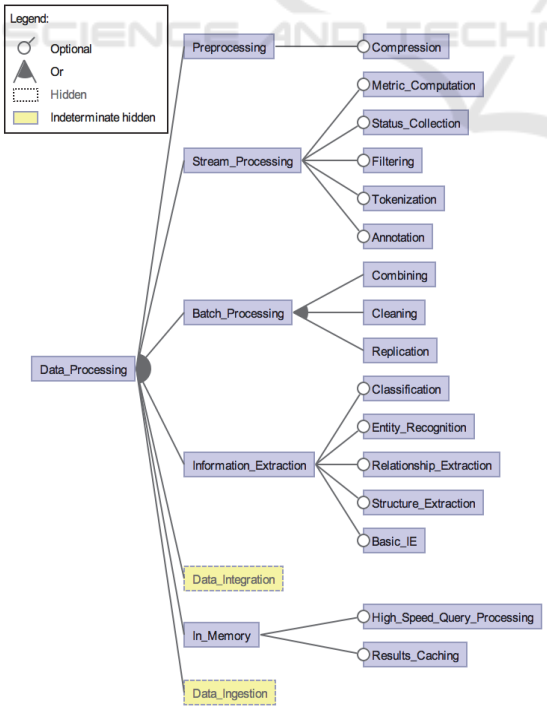

Figure 11: Feature diagram of Data Processing feature of Twitter.

the Twitter's software architecture. Moreover, this architecture is also capable of doing sophisticated ad hoc analyses that are designed to answer some specific questions.

As a result, we observed that while all the top level features are covered by the two systems' (Facebook & Twitter) archicecture, the big data security related features could not be derived from the architectural descriptions of the system.

4 CONCLUSIONS

In this study we have derive a feature model for Big Data systems using a systematic domain analysis process. Based on selected relevant papers we have been able to derive both the common and variant features of Big Data systems and represent this as a feature diagram. We have discussed the features separately and illustrated the adoption of the feature model for characterizing parts of two different systems including Facebook and Twitter. The feature diagram provides a first initial insight in the overall configuration space of Big Data systems. As a future work, we plan to illustrate our approach for deriving the Big Data architectures.

REFERENCES

Araùjo, J., Baniassad, E., Clements, P., Moreira, A., Rashid, A., & Tekinerdogan, B., 2005. *Early aspects: The current landscape*. Technical Notes, CMU/SEI and Lancaster University.

Arrango, G., 1994. *Domain Analysis Methods in Software Reusability*. Schäfer, R. Prieto-Díaz, and M. Matsumoto (Eds.), Ellis Horwood, New York, New York, pp. 17-49.

Ballard, C., Compert, C., Jesionowski, T., Milman, I., Plants, B., Rosen, B., & Smith, H., 2014. *Information Governance Principles and Practices for a Big Data Landscape*, IBM Redbooks.

Chapelle, D., 2013. *Big Data & Analytics Reference Architecture*, An Oracle White Paper.

Czarnecki, K., Hwan, C., Kim, P., & Kalleberg, K. T., 2006. Feature models are views on ontologies. In *Software Product Line Conference, 2006 10th International (pp. 41-51)*. IEEE.

Geerdink, B., 2013. A reference architecture for big data solutions introducing a model to perform predictive analytics using big data technology. In *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for (pp. 71-76)*. IEEE.

Harsu, M., 2002. *A survey on domain engineering*. Tampere University of Technology.

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., &

Peterson, A. S., 1990. *Feature-oriented domain analysis (FODA) feasibility study (No. CMU/SEI-90-TR-21)*. Carnegie-Mellon Univ. Pittsburgh Pa. Software Engineering Inst.

Laney, D., 2001. *3D Data Management: Controlling Data Volume, Velocity and Variety*. Meta-Group Report #949.

Lee, K., Kang, K. C., & Lee, J. (2002). *Concepts and guidelines of feature modeling for product line software engineering.* In Software Reuse: Methods, Techniques, and Tools (pp. 62-77). Springer Berlin Heidelberg.

Maier, M., Serebrenik, A., & Vanderfeesten, I. T. P., 2013. *Towards a Big Data Reference Architecture*.

Marz, N., & Warren, J., 2015. *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*, Manning Publications Co..

May, W., 2014. *Draft NIST Big Data Interoperability Framework: Volume 6 Reference Architecture*.

Mishne, G., Dalton, J., Li, Z., Sharma, A., & Lin, J. 2013. Fast data in the era of big data: Twitter's real-time related query suggestion architecture. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (pp. 1147-1158)*. ACM.

Oracle, 2013. *Information Management and Big Data A Reference Architecture*, An Oracle White Paper.

Pääkkönen, P., & Pakkala, D., 2015. *Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems*. Big Data Research.

Sinnema, M., & Deelstra, S., 2007. *Classifying variability modeling techniques*. Information and Software Technology, 49(7), 717-739.

Soares, S., 2012. *Big Data Governance*. Information Asset, LLC.

Tekinerdogan, B., & Akşit, M., 2001. *Classifying and Evaluating Architecture Design Methods, in Software Architectures and Component Technology: The State of the Art in Research and Practice*. M. Aksit (Ed.), Boston:Kluwer Academic Publishers, pp. 3 - 27.

Tekinerdogan, B., Bilir, S., & Abatlevi, C. (2005). Integrating platform selection rules in the model driven architecture approach. In *Model Driven Architecture(pp. 159-173)*. Springer Berlin Heidelberg.

Tekinerdogan, B., & Öztürk, K. (2013). Feature-Driven Design of SaaS Architectures. In *Software Engineering Frameworks for the Cloud Computing Paradigm (pp. 189-212)*. Springer London.

Thusoo, A., Shao, Z., Anthony, S., Borthakur, D., Jain, N., Sen Sarma, J., Murthy, R. and Liu, H., 2010, June. Data warehousing and analytics infrastructure at facebook. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (pp. 1013-1020)*. ACM.