Chapter 9
Data Pipelines: Modeling and Evaluation of Models

Kaïs Chaabouni and Alessandra Bagnato

Abstract This chapter outlines the utility of data pipelines modeling in the context of a data driven project and enumerates metrics for evaluating the quality of the data modeling regarding the readability and the comprehensibility of the models. We start with explaining the challenges surrounding the DataBio project that led to the adoption of data pipelines modeling using the Enterprise Architecture language ArchiMate. Then we present the data modeling process with examples from DataBio pilot studies starting with modeling software components provided by project stakeholders and ending up with integration of components into data pipelines that achieve the data analytics lifecycle intended by the pilot study. We end the chapter with the evaluation of the quality of DataBio data pipelines models with metrics collected by a monitoring tool for ArchiMate models.

9.1 Introduction

DataBio [1] aims to develop a platform that exploits the potential of big data technologies in the domains of agriculture, fishery and forestry. Given the complexity of the task, the project decided to adopt the "Enterprise Architecture" modelling language "ArchiMate 3.0" [2, 3] as a common modelling framework for representing the requirements of the pilots and modelling the technical architecture of the components, thus facilitating communication and comprehension among partners. Most of the software components interact with data from different origins and with various formats such as satellite imagery, sensors data, geospatial data (see Chap. 4), etc. In each pilot, components are connected together through several interfaces to form a data pipeline, (see Chap. 1) in which each component has a specific function in the

K. Chaabouni (✉) · A. Bagnato (✉)
Softeam Research Department, 21 Avenue Victor Hugo, 75016 Paris, France
e-mail: kais.chaabouni@softeam.fr

A. Bagnato
e-mail: alessandra.bagnato@softeam.fr

C. Södergård et al. (eds.), *Big Data in Bioeconomy*,
https://doi.org/10.1007/978-3-030-71069-9_9

data value chain such as data collecting, data processing, data analytics and visualization. The modelling approach consists of representing the components and the data pipelines according to a predefined model template. The modelling environment used for this task is "Modelio" [4], which allows contributors to collaborate around a synchronized ArchiMate model. The collaboration around the models faces some challenges regarding their potential to be efficiently exploited. Hence, we define metrics for evaluating the quality of the models and we measure continuously the quality level according to these metrics using a monitoring platform.

9.2 Modelling Data Pipelines

The Enterprise Architecture language ArchiMate provides several concepts for modelling the different layers of the enterprise:

- The physical layer contains the devices and their connections, which are used in the deployment of the IT system.
- The application layer contains the software services and the data flow.
- The business layer contains business services, interfaces and actors.

The modelling of software components enabled the DataBio partners working in the various pilots to easily understand the underlying functioning of each pilot. At first, partners were asked to provide models for the software components that they have provided. In a second time round, the partners were instructed to provide data pipelines diagrams that highlight the integration of the components in each pilot study. All of the software components, pipelines and datasets can be found at the DataBio Hub [5].

9.2.1 Modelling Software Components

The project developed a naming convention, where each software component has an identifier with the pattern "Cxx.yy" where "C" refers to the word "Component", "xx" represents the number of the partner that had provided the component and "yy" represents the component number of that partner. Datasets are correspondingly expressed with the template "Dxx.yy", For example, "C16.01" denotes the first component from partner 16, which is VTT .An expanded notation is "C16.01: OpenVA (VTT)", as the component is called OpenVA, which is a platform that consists of software modules that are used as building blocks of web based visualisation and analytics applications [6]. Components are modelled with diagrams that follow a predefined template. These diagrams include deployment view, interfaces view and subordinates view.

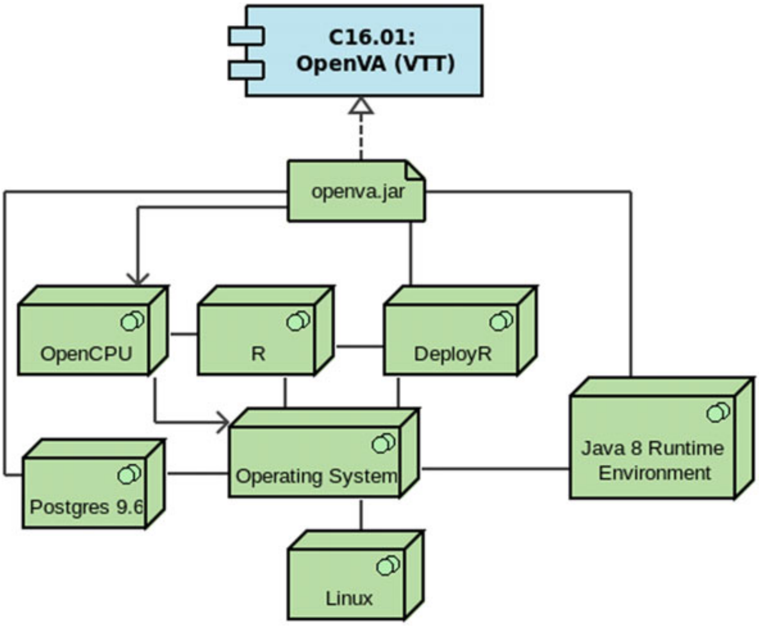

Fig. 9.1 OpenVA deployment view

9.2.1.1 Deployment View

The deployment view describes how the application is being deployed by representing the executables of the software component, the software dependencies and the physical environment required for running the application. Figure 9.1 shows an example of the deployment view of the component "C16.01: OpenVA (VTT)". As shown by the figure, OpenVA is packaged as JAR Java Package (openva.jar) which is run as a server via Java Runtime Environment (JRE). The database is handled by the Database Management System (DBMS) PostgreSQL 9.6. OpenVA server depends on two applications: DeployR and OpenCPU. DeployR is an open source application that turns R scripts into web services, so R code can be executed by applications running on a secure server. The OpenCPU server provides an HTTP API for data analysis for running R scripts on the server. OpenCPU uses standard R packaging to deploy server applications.

9.2.1.2 Subordinates View

The Subordinates view describes the subcomponents of the component such as the libraries, modules and frameworks that compose the whole application. For example, Fig. 9.2 shows the subcomponents of "C16.01: OpenVA (VTT)" which is composed

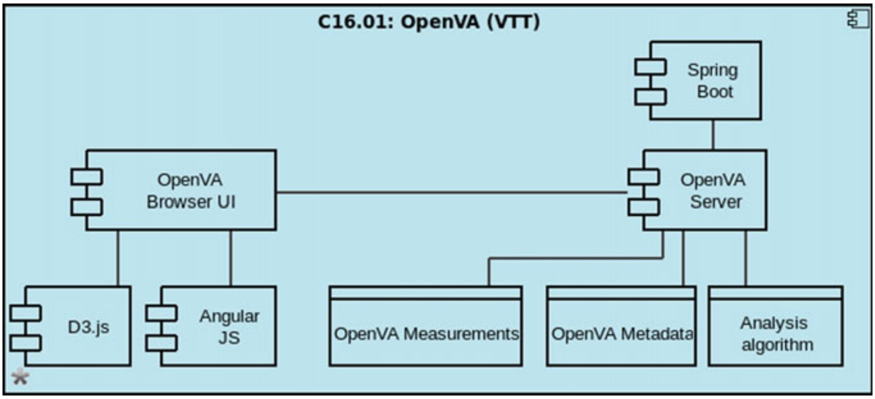

Fig. 9.2 OpenVA Subordinates view

of "OpenVA server" (the backend of the application) and "OpenVA Browser UI" (the frontend of the application).

9.2.1.3 Interfaces View

The interface view shows the provided and required interfaces of components which are designed for interactions with users or with other components through various communication protocols [7]. Figure 9.3 shows an example of the interface view of the component "C16.01: OpenVA (VTT)", which offers a web user interface for accessing OpenVA via a browser. OpenVA is also accessible via interfaces that can be provided by other components such as JDBC interface for accessing OpenVA database, Sqoop export tool for moving a set of files from HDFS (Hadoop Distributed File System) to RDBMS (Relational DataBase Management System).

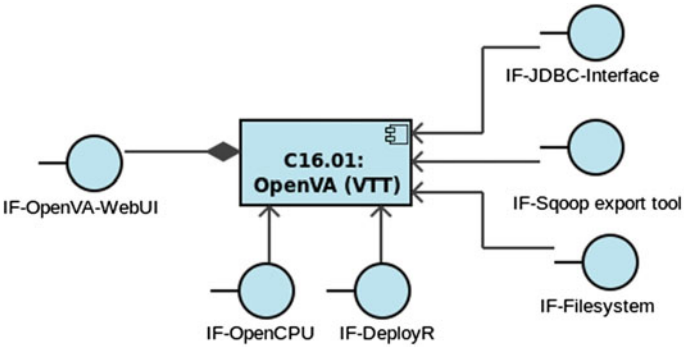

Fig. 9.3 OpenVA interface view

9.2.2 Integrating Components into Data Pipelines

Each pilot integrates in its workflow a set of software components that interact with each other in order to process huge amounts of heterogeneous data. These sets of interoperable software components are called pipelines and work as so-called white boxes showing the internal wiring and data flow between the single components of the pipeline. Hence, we model these pipelines with a "Pipeline View" that shows the different connections between components and a "LifeCycle View" that emphasizes the data value chain.

9.2.2.1 Pipeline View

Pipeline Views illustrate the connections between the different components and the interfaces that allow them to interact together. Figure 9.4 illustrates the Pipeline View of the fishery pilot "Oceanic tuna fisheries immediate operational choices" [8]. In this pilot, measurements from the ship engines are recorded continuously and are then uploaded to the ship owner server. These measurements are processed and analysed by three major components: "C16.01: OpenVA (VTT)", "C34.01: EXUS Analytics Framework (EXUS)" and "C19.01: Proton (IBM)". Each of these components offers a web interface for interacting with users and visualizing data via dashboards. "C19.01: Proton" is an event processing engine that processes events from different sources such as reading from files or from RESTful API. In this example, we receive sensor readings from the ship's monitoring and logging system which are

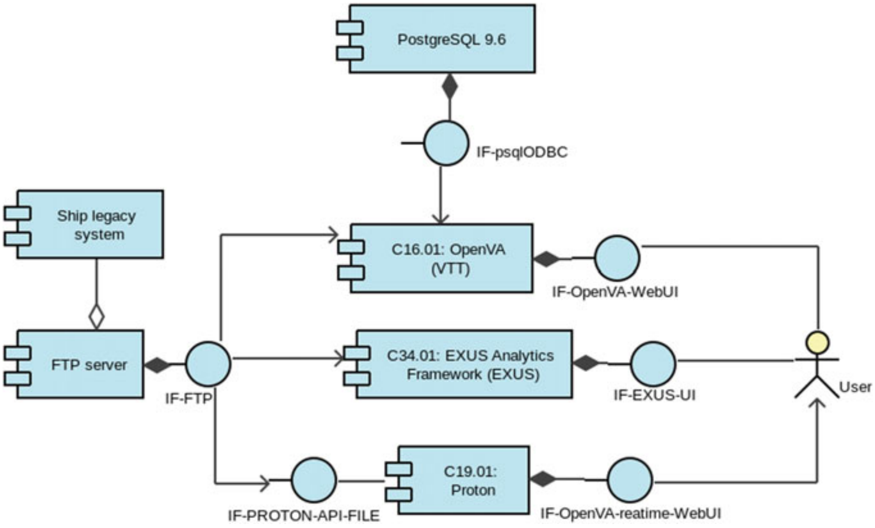

Fig. 9.4 "Oceanic tuna fisheries immediate operational choices" pilot—Pipeline view

then stored in the file system via FTP, from which it is read by Proton's file adapter and streamed into Proton engine for processing.

9.2.2.2 Lifecycle View

The lifecycle view shows the different tasks accomplished by each component along the data value chain according to the Big Data Value Reference Model [6, 9]. Figure 9.5 illustrates the Lifecycle View of the same fishery pilot as above "Oceanic tuna fisheries immediate operational choices". In this figure, we can see that the "Ship legacy system" is responsible for collecting raw sensor data. Then, custom tools and specific scripts are applied for data preparing (cleaning and transforming data) before executing the analytics tools. Finally the three major tools "C16.01: OpenVA (VTT)", "C34.01: EXUS Analytics Framework (EXUS)" and "C19.01: Proton" are used for data analytics and data visualisation.

9.3 Models Quality Metrics

The DataBio ArchiMate models are structured in five so-called projects: three projects for describing the pilots of agriculture, forestry and fishery, one project for modelling software and IoT system components and one project for modelling

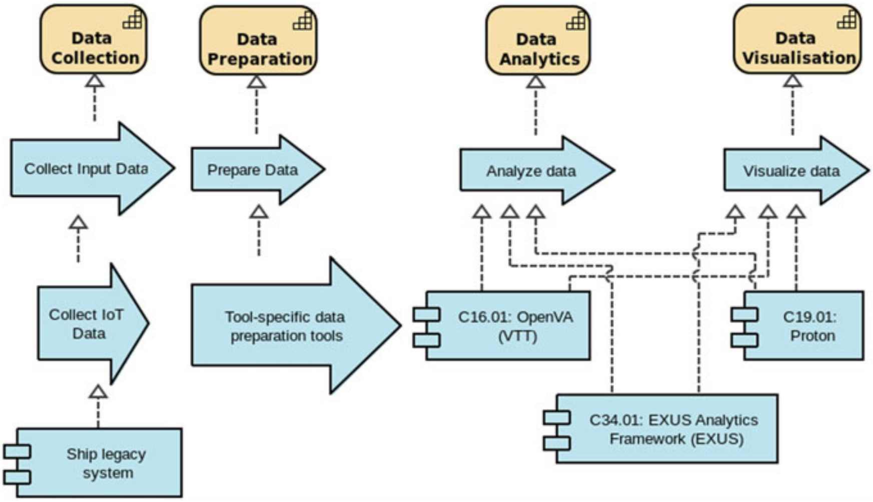

Fig. 9.5 "Oceanic tuna fisheries immediate operational choices" pilot—Lifecycle View

“Earth Observation” data services. These projects are monitored by “Measure Platform” [10], which is a monitoring platform that allows to collect periodic measurements on monitored projects. In this case, these measurements are obtained via the model indexing tool “Hawk” [11], which processes the queries of ArchiMate models. For each metric, we define a query for Hawk to interrogate from the models. After this, we store and visualize the collected measurement via the Measure Platform [12].

9.3.1 Metrics for the Quality of the Modelling with Modelio

Ensuring a better quality of the models begins with monitoring the modelling process with Modelio, which follows the creation of elements, folders, diagrams and documentation inside an ArchiMate project. We present here metrics that reflect how optimal the usage of Modelio is to guarantee a complete system design.

9.3.1.1 Percentage of Unused Elements in Diagrams

“Unused elements” are elements that have not been represented in diagrams and therefore do not bring any added value to the final generated diagrams. Each Modelio project contains a “Model Explorer” that is divided into two types of directories; one directory for managing the created elements and one directory for managing the diagrams. In the first directory we can visualize the list of all the elements that are created in the project whether they are displayed in diagrams or not. The second directory is for managing the diagrams that represent elements and their relationships. The “percentage of unused elements” metric could be an indicator of an incomplete modelling, where the element was created, but its relation with the rest of elements has not been yet specified. The unused elements could also be explained by the fact that users of Modelio sometimes create elements in diagrams and then mask them from the diagrams without deleting them from the project’s Model Explorer. Moreover, this metric could also be an indicator of inefficiency, because it points to the incomplete work and to the wasted amount of work for creating useless elements. In addition, the unused elements will unnecessarily extend the list of displayed elements inside the Model Explorer, which would complicate the navigation for the user. Figure 9.6 shows that the percentage of unused elements in the monitored ArchiMate repositories in DataBio sub-projects has been between 20 and 50%.

9.3.1.2 Percentage of Duplicate Elements

The presence of duplicate elements in the models adds complexity for Modelio users as the redundancies complicate needlessly the visibility of the project and cause confusion, when choosing a suitable element. Moreover, the duplication of elements

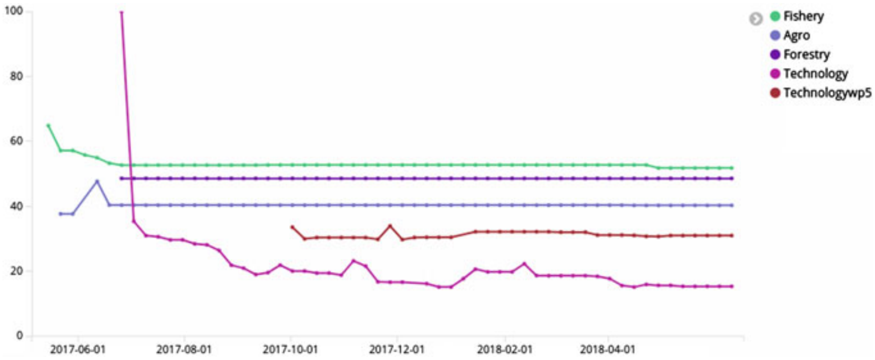

Fig. 9.6 Percentage of unused ArchiMate elements in diagrams

prevents the full exploitation of Modelio features such as identifying shared elements. Those elements are represented by several diagrams or by all the relations associated to the specified element.

9.3.1.3 Percentage of Empty Diagrams

The presence of empty diagrams is an indicator of unfinished or obsolete diagrams that need to be removed or updated.

9.3.1.4 Frequency of SVN Commits

The Modelio projects are stored as SVN (Subversion, an open-source version control system) repositories and can therefore be monitored by observing the frequency of updates of the models. The number of SVN commits per week shows the periods of time during which the work on models has been carried out. This metric does not reflect the real amount of the committed work, but rather the frequency of submitting new releases of the monitored models. Figure 9.7 shows the number of weekly SVN commits in DataBio Archimate Models. We can see from this figure that the major work on the models has been done between May 2017 and June 2018. Other SVN related measures could be conducted such as the number of contributors and the frequency of submitting new updates by each contributor.

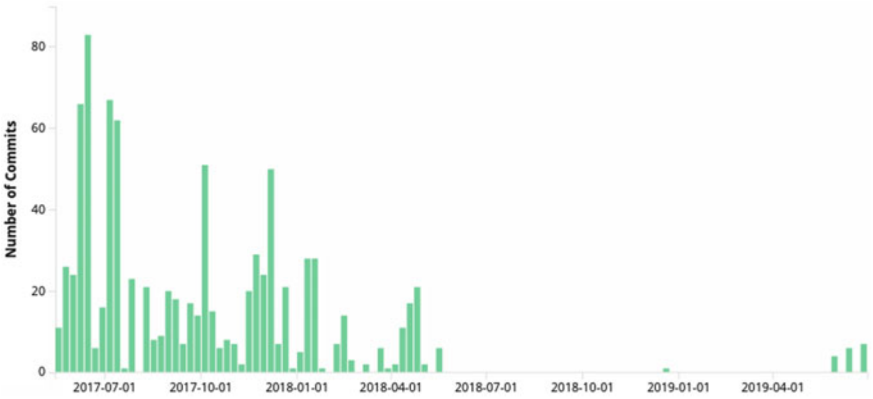

Fig. 9.7 Number of commits per week

9.3.2 ArchiMate Comprehensibility Metrics

The quality evaluation of ArchiMate views is based on several criteria that capture how well the views have fulfilled their purpose, especially their ability to help understand certain aspects in the project. Therefore, we introduce the comprehensibility metrics that evaluate how easy it is for the user to read the diagram and how easy it is to understand the model. The readability of the diagrams is impacted by how easy it is to read elements in diagrams, distinguish them from each other and find all the links between them. The understandability of the model from the provided diagrams depends on how easy it is to understand the whole organisation, the purpose of each component, service or process and the interactions between them.

9.3.2.1 Average Number of Elements per Diagram

The average number of elements per diagram shows how easy it is to read a diagram. Having a large number of elements in the same diagram will result in a dense diagram or in tiny elements inside the diagram, if it is scaled to a page or screen size. This makes it harder for users to read. On the other hand, having a very low number of elements per diagram could reflect a very fragmented model. We recommend between 8 and 25 elements per diagram, which is the case in the DataBio projects (see Fig. 9.8).

9.3.2.2 Average Number of Relationships per Element

The average number of relationships per element reflects the congestion of associations between elements and directly affects the readability of the diagram.

This number should be between 1 and 4 relationships per element. A Relationships/Elements ratio approaching 0 indicates that there are very few connections between the elements in the diagrams. On the other hand, a Relationships/Elements ratio exceeding 4 could indicate a big density of connections in the diagrams.

9.3.2.3 Documentation Size per Element

One key factor for understanding diagram elements is a documentation that provides definitions and comments about the elements and how they are used in the project. This metric evaluates the average size of the textual description provided for an element. This could be considered as an indicator of how detailed the description of the element is. Figure 9.9 shows the history of measured documentation size (number of words) per element in the monitored projects, which have an acceptable average size. However, this measure does not show the disparity of documentation, where

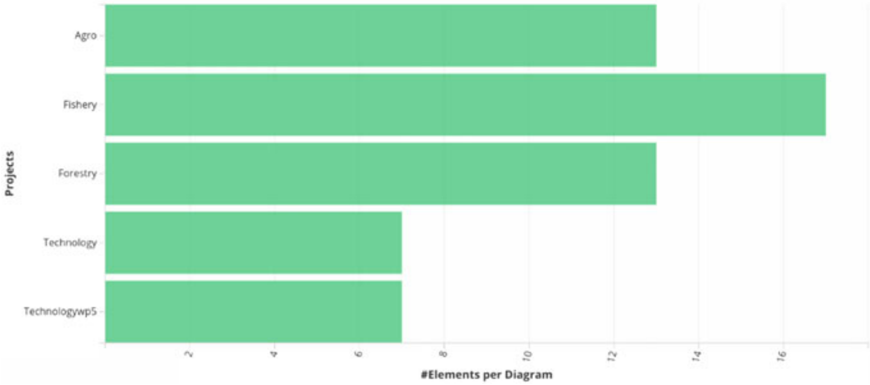

Fig. 9.8 Average number of elements per diagram

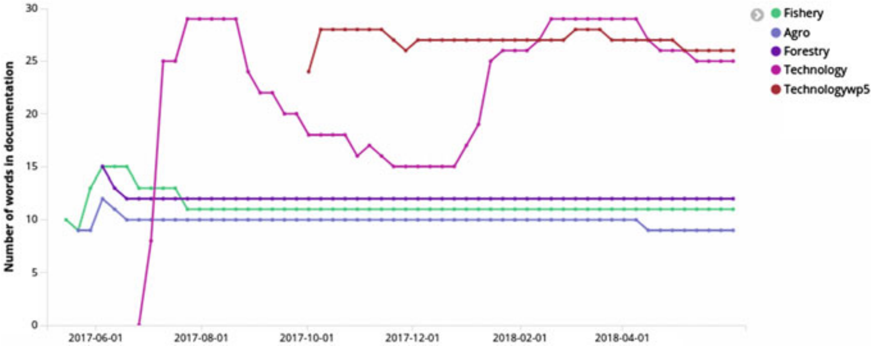

Fig. 9.9 Documentation size per element

some elements are described with big paragraphs and others have no description at all.

9.3.2.4 Documentation Size per Diagram

This metric evaluates the understandability of diagrams by measuring the documentation size diagram. It is similar to the previous one with the difference that it calculates the documentation size per diagram instead of the documentation size per element. This allows us to locate in more detail the diagrams that are lacking description.

9.3.2.5 Percentage of Documented Elements

This metric focuses on the documented part of the models. It measures the percentage of the documented elements. Apart from the self-evident elements, which are understandable just by name, it is highly recommended to describe the remaining elements, especially the elements containing abbreviations, which are not well known to everyone. Figure 9.10 shows the percentage of the documented elements in the monitored projects. The projects, which describe the agro, fishery and forestry pilots, have few documented elements (between 15% and 24%). This is explained by the clear and detailed namings of the motivation and strategy elements, which therefore do not require further explanations. On the other hand, the technology projects deal with a lot of technological components that require documentation. Hence, the documented elements represent more than 58% of the total elements in these projects.

9.3.3 Metrics for Model's Size

The model size is an indicator of the modelling progress as it reflects the number of created diagrams and elements and their relationships inside diagrams. The model size is also an indicator of the complexity of the model. The number of non-empty diagrams reflect the actual number of the models existing in the studied organisation. In our case, the ArchiMate models contain more than 500 non-empty diagrams. This makes it more complex to understand the whole project.

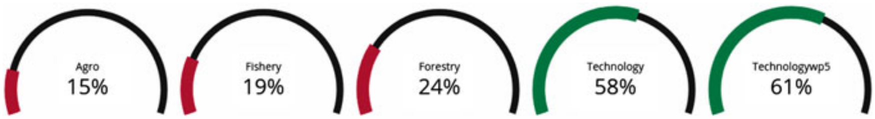

Fig. 9.10 Percentage of documented elements per project

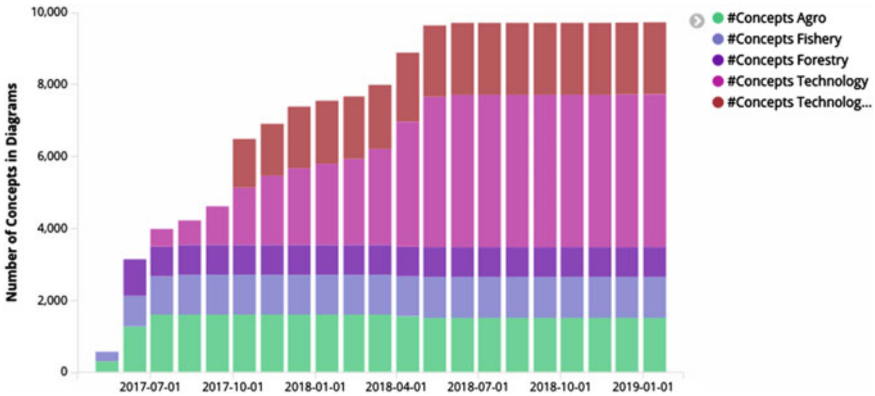

Fig. 9.11 Number of concepts represented in diagrams

9.3.3.1 Total Number of ArchiMate Concepts Used in Diagrams

Since diagrams differ in size, the number of overall ArchiMate concepts used in diagrams add information about the size of the models. The ArchiMate concepts considered here contain the elements represented in the diagrams and the relationships between the elements. Figure 9.11 shows the evolution of the total number of ArchiMate concepts and the proportion of concepts in each DataBio project. We can see that the total number of ArchiMate concepts is very close to 10000 elements, which is an indicator of the complexity of the project.

9.4 Conclusion and Future Vision

The modelling of DataBio components and data pipelines provided more clarity to the project and helped to understand the architecture of the used software components and their integration in the pilots workflows. Moreover, the created models have also contributed to the process of requirements elicitation throughout the project period and to the efficient writing of the documentation. In order to monitor the quality of the models, we have defined a metric that evaluates the efficiency of the modelling process, the comprehensibility of the models and the model size. The metric discussed here could be applied also in other projects, where the modelling tool Modelio or the modelling language ArchiMate are in use [12]. The proposed metric indicates that the quality level in DataBio is acceptable as comes to the efficiency of the modelling process and the comprehensibility of the models. However, we note that there are some areas to be improved such as the cohesion and the completeness of the models. The analysis showed that the models are lacking a more holistic view of the DataBio project, where there is a big data platform or environment offering services and components to the different pilots. Hence, we aim at finding more metrics for evaluating the cohesion of the models and expressing the interdependency between elements and diagrams inside the project. Moreover, our analysis showed that there are many incomplete and undetailed diagrams and we need therefore a metric that expresses the completeness and the maturity of the diagrams.

Acknowledgements The authors would like to thank Antonio Garcia-Dominguez from School of Engineering and Applied Science at Aston University, Birmingham, UK for his support with the Hawk tool. We would like also to acknowledge the effort of ITEA3 project 14009, MEASURE Project and thank their consortium team members for their feedback on this work. Furthermore, the research leading to these results has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 732064, Databio Project.

References

1. DataBio Website. (2019). https://www.databio.eu/en/. Last accessed May 14, 2019.
2. Desfray, P., & Gilbert R. (2018). TOGAF, Archimate, UML et BPMN-3e éd. Dunod.
3. Fritscher, B., & Pigneur, Y. (2011). Business IT alignment from business model to enterprise architecture. In *International conference on advanced information systems engineering* (pp. 4–15). Springer.
4. Modeliosoft—Modelio BA Archimate Enterprise Architect. (2019). https://www.modeliosoft.com/en/products/modelio-ba-archimate-enterprise-architect.html (2019/04/18).
5. DataBioHub Website. (2019). https://www.databiohub.eu/. Last accessed May 14, 2019.
6. DataBio public deliverable D4.1 Platform and Interfaces. (2019). https://www.databio.eu/wp-content/uploads/2017/05/DataBio_D4.1-Platform-and-Interfaces_v1.0_2018-05-31_VTT.pdf. Last accessed Aug 13, 2019.
7. Chaabouni, K., Bagnato, A., Walderhaug, S., & Berre, A. J. Södergård, C., Sadovykh, A. (2019). *Enterprise architecture modelling with ArchiMate* (pp. 79–84). STAF (Co-Located Events).
8. DataBio public deliverable D4.2 Services for Tests (Public version). (2019). https://www.databio.eu/wp-content/uploads/2017/05/DataBio_D4.2-Services-for-Tests_public-version.pdf. Last accessed Aug 14, 2019.
9. BDVA Strategic Research and Innovation Agenda, version 4.0, October 2017. (2019). http://www.bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf. Last accessed Aug 14, 2019.
10. Measure Platform. (2019). http://measure-platform.org/. Last accessed May 21, 2019.
11. Hawk Tool. (2019). https://github.com/mondo-project/mondo-hawk. Last accessed May 21, 2019.
12. Chaabouni, K., Bagnato, A. (2019). *Antonio García-Domínguez: monitoring archimate models for databio project* (pp 583–589). PROFES.