MSc Computer Science
Final Project

Adapted CRISP-DM approach for recommendation system development for most suitable open-source ETL tool

Jurgen Grotentraast

Supervisor: Faiza Bukhsh, Nacir Bouali, João Rebelo Moreira, Luuk Peters, Ronald van Aalderen

December, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Abstract

Implementing data integration or Extract-Transform-Load (ETL) workflows is difficult because of the many different factors that play a role. Choosing the right tool for this implementation is therefore vital to ensure the developers' preferences and requirements are met. However, finding this tool is just as complex because different tools have different strengths, weaknesses, and capabilities that need to be considered. This paper covers the adaptation of the CRoss Industry Standard Process for Data Mining (CRISP-DM) methodology for developing a recommendation system for open-source ETL tools. This recommendation system helps users find a suitable ETL tool for their use case. Therefore, interviews were conducted with developers to find the key aspects of ETL tools among others security, ETL pipeline design, and hosting. Different open-source tools were analyzed on these key aspects in the form of an aspect matrix. This aspect matrix was transformed into a scorecard to filter and rank different tools based on requirements. A questionnaire was created to gather the requirements and provide recommendations to the user. Lastly, the recommendation system was evaluated in three ways. As a form of self-reflection compliance with the seven guidelines by Hevner et al. was rated to reflect on the development process. The recommendation system was evaluated with a survey in which participants could use it to rate its understandability, usability, and clarity. Overall, participants rated the recommendation system a 6.8 out of 10. The main improvements could be made in the presentation and motivation of the recommendations. The results indicate the adaptations made to the CRISP-DM methodology were appropriate and useable for developing a recommendation system.

Keywords: ETL tools, CRISP-DM, Recommendation system

Contents

Chapter 1

Introduction

With today's world's still-growing value of data, many organizations have invested in developing a data warehouse (DW). A DW stores data differently to efficiently analyze business data [26]. DWs can be used for analyzing and improving business processes [39], but also to get a better understanding of for example the financial situation of an organization [33]. A DW utilizes historical data to show trends, averages, and bottlenecks in a process or production chain and to show what areas of this process or production chain can be improved [10]. Published research papers in this area focus on improving or developing design techniques [2, 3]; development of a DW for a specific use case; and creating new concepts such as the data lake or data lakehouse [4, 20, 31, 38].

A DW captures data from one or multiple sources, transforms the data in such a way that aggregations on this data are easy and fast to execute, and finally loads this data into the DW database. This process is called extract-transform-load (ETL). Over the years many tools and software solutions have been developed to aid people in this process. Some tools are purely programming libraries or extensions that help the user to achieve what they want [6, 30, 44], whereas other software applications are developed further such that they can be used to build ETL pipelines with minimal coding. Companies like Amazon, Microsoft, and Google have developed cloud-based software applications for creating DW solutions. However, these are often costly and require a subscription to their entire cloud platform to use them [1, 23, 34].

Fortunately, over the last couple of years, open-source ETL tools, such as Apache Airflow, Prefect, and Dagster, have been developed further and further [35]. This means that open-source tools now have the same functionality as expensive enterprise solutions. Furthermore, these open-source tools allow the user to build upon the tool themselves if something is missing. For example, if a connection to a specific source of data is not yet part of the tool, the user can build a custom connector through an API and still extract all the data they want.

Topicus .Finance is an IT company based in the Netherlands and Vietnam and part of the larger Topicus brand name. Alongside Finance, Topicus has divisions working in Education, Health care, and the social domain. Within each of these divisions, Topicus has developed several applications. This study was conducted with the help of Topicus .Finance who had chosen an open-source ETL tool is now regretting that choice. They wish to replace their ETL tool, however, they are unsure which open-source tool would be suitable for them.

1.1 Problem statement

Within Topicus, not only the Finance division was facing the choice of a new ETL tool. However, with the amount of open-source ETL tools available, finding the one that is right for the task at hand is difficult [21, 37]. Earlier, a tool was often chosen without knowing if it was suitable for the task at hand [21].

Nowadays, guidelines exist for choosing the right tool [21, 43]. However, these methods first require the user to identify possible tools themselves which can lead to viable options being missed. Furthermore, there are different methodologies extensively outlined for developing new software or models that fit the requirements of a specific task, for example, a waterfall or agile approach for software development [21] and the CRoss Industry Standard Process for Data Mining (CRISP-DM) and the Sample, Explore, Modify, Model, and Assess (SEMMA) methodologies for model development.

This paper modifies the CRISP-DM methodology to develop a recommendation system for ETL tools. This methodology was chosen because it ensures a proper understanding of the problem to develop a solution. Furthermore, the cyclic approach allows for the improvement of the results in each iteration. This adapted methodology was then used to develop an ETL recommendation system that should recommend the most suitable tool for a specific use case.

1.2 Research questions

The problem statement mentioned above leads to the following research question:

How can an adapted CRISP-DM methodology be used to develop a recommendation system for open-source ETL tools?

This question can be broken up into the following three sub-questions:

- **Sub-RQ1:** What are the key aspects of an ETL tool for a specific use case?
- **Sub-RQ2:** How do different open-source ETL tools handle these key aspects?
- **Sub-RQ3:** How can recommendations for open-source ETL tools be determined based on requirements?
- **Sub-RQ4:** How useful do users find the recommendations for open-source ETL tools?
- **Sub-RQ5:** Does the adapted CRISP-DM approach result in a working recommendation system?

The answers from each sub-question will help in answering the main research question.

1.3 Thesis structure

This paper continues by discussing the background in chapter 2. Chapter 3 covers the adaptation made to the CRISP-DM methodology and each phase of this adapted methodology used to design and evaluate the created recommendation system. Chapter 4 shows the design of the recommendation system and chapter 5 shows the results of the recommendation system's evaluation. Chapter 6 discusses the results and their implications. Finally, in chapter 7 a conclusion is drawn by answering the research questions, and the limitations and potential future work is discussed.

Chapter 2

Background

The following paragraphs focus on related research that was found. The findings of related literature are briefly summarized and we discuss how these findings complement this study's results. Furthermore, two systematic literature reviews were performed by the researchers before the start of this study which is also briefly discussed.

One study on open-source ETL tools by Biswas et al.[6] was found, which compares different Python libraries that offer ETL capabilities. Several of the tools mentioned in the paper by Biswal et al. were also used as part of the recommendation system created in this study. These tools were extended in one of the literature studies of the previous study, examined in greater detail, and used as part of the recommendation system as possible suggestions [25].

Several studies that focus on the trends in data warehousing were found. First, there were studies published in 2018 reviewing trends up to and including 2017 [9, 11, 22, 32]. While each study had its take and focus, all these studies recognized the increase in data volume which resulted in a shift from traditional data warehousing to big data warehousing. Furthermore, these studies showed that the architecture of a DW has also shifted over the years from a DW to data lakes, to lakehouses, and now to data meshes. Where the standard used to be a relational database with a clear structure, these studies show that the architectures up until 2018 also started to shift to incorporate more NoSQL capabilities as the data that these systems had to handle became more and more unstructured. Moreover, the designing of a DW also shows several clear approaches that have emerged over the past years. The approaches were classified into five categories:

1. **data-driven:** which starts the design phase by analyzing the source data
2. **requirement-driven:** which starts at the other end, looking at the requirements from the end user
3. **mixed:** which combine a data-driven and requirement-driven approach
4. **query-based:** which start by defining the workload the DW should take care of
5. **pattern-based:** which also starts at the source data but looks for multidimensional patterns

These studies also show that a DW has to handle more and more types of data from different sources and should therefore be interoperable with as many systems as possible. The studies that were found show trends and approaches up until 2018, these were extended in the second literature study done before this study, which looked at the trends from 2018 up until 2024. The results showed whether trends that started six or seven years ago are still relevant, and which completely new trends have emerged.

S. Eom published a study on the current state and emerging trends regarding decision support systems, business intelligence, and data analysis [16]. These kinds of systems are often based on a DW, and therefore, trends in these systems might affect trends in data warehousing. The study by Eom focuses on the direction of research on decision support systems, data analytics systems, and business intelligence systems, as well as use cases of these kinds of systems.

Dhaouadi et al. published a work on the classical approach and new trends in the design of the ETL process [14]. Dhaouadi et al. identified the following six classes on ETL modeling approaches.

1. UML
2. Ontology
3. Model Driven Architecture
4. Graphical Flow formalism (BPMN, CPN, YAWL, data visualization flow)
5. Ad hoc formalisms (conceptual constructs, CommonCube, EMD)
6. Big data approaches

The conclusion of Dhaouadi et al. shows that ETL process modeling based on standard modeling languages like UML or BPMN were confirmed to be powerful methods as they standardize the ETL workflow design. ETL process modeling based on ontologies showed an easy identification of the schema of the data sources and DW. Furthermore, ontologies are most suitable for capturing the semantics of the domain model. However, mapping between different sources was considered an extremely complex task.

Next, one advantage of model-driven architecture (MDA) based process modeling was separating business logic and technology by providing different layers that lead to interoperable, reusable, and portable software components and data models. The biggest advantage of these MDA-based methods was the automated transformations of models to implementations, which are done through automatic code generation from these models. One drawback of these automated transformations is the reliance on patterns and references to constantly updated libraries.

The use of patterns also showed interesting results, as patterns allow for reusability of parts of the ETL process, reducing potential design errors in future parts. The work by Dhaouadi et al. is a well-suited addition to the results found in this study. The focus of Dhaouadi et al. highlights the different approaches of a sub-area of DW research that can be interesting as part of the key aspects.

As mentioned, a study was conducted before the start of this one. This prior study was conducted as a preparatory study for this research. That study consisted of two systematic literature studies. First, we conducted a literature study on open-source ETL tools to find as many open-source ETL tools currently available that were last updated in or after 2023. This took the term ETL tools in its broadest sense to include as many relevant applications as possible. As this did not lead to a complete list, this part was extended with results found through Google. The second part of this prior study was another literature study on the trends and approaches in the research, design, development, implementation, and improvement of a DW from 2018 up until 2024. These trends were used to create the interview questions and influenced the key aspects that were found. The results are briefly recapped in appendix A. The full study is available on Github [25].

Chapter 3

Methodology

The following sections explore how the research questions are answered. As mentioned in the introduction of this paper, an adaption of the CRISP-DM methodology was used to develop a suggestion tool for the most suitable ETL tool.

3.1 CRISP-DM

CRISP-DM is a design method published in 1999 and was meant as a standard for data mining processes across domains [28, 46]. CRISP-DM consists of six phases. Each phase has its own goal for implementing a data mining model. CRISP-DM mainly focuses on data mining and model development such as machine learning models. This means it is not entirely one-on-one applicable to the research presented in this paper. However, the different phases can be adapted to make them directly applicable. A high-level comparison of the original CRISP-DM and the adapted form used in this study is displayed in figure 3.1.

The two biggest differences are, first, having the deployment before the evaluation as this allows the evaluation can be performed on the working recommendation system. Second, the data understanding and data preparation phases are usually two separate phases, however, these are combined into one and consist of gathering information on the considered tools and transforming this information into a usable format. Further adaptations have been made to each phase individually to reflect the development of a recommendation system. These adaptations are further discussed in the following sections.

A complete overview of the employed method and their respective outcomes is presented in figure 3.2. Each phase helps answer one or two sub-RQs outlined in section 1.2. Sub-RQ 1 is answered with the results of the business understanding phase in the form of a list of key aspects. The data understanding and preparation phase results help answer sub-RQ 2 in the form of a scorecard. The modeling phase helps answer sub-RQ 3, as the logic for recommendations is finished in this phase. Sub-RQs 4 and 5 are answered with the evaluation phase. Each phase is discussed in more detail in the sections below.

3.2 Business understanding

The first phase of CRISP-DM is the business understanding phase. This phase focuses on understanding the problem and objectives that we wish to solve [28, 46]. Therefore, to design a recommendation system for choosing the right ETL tool for a specific use case, it is important to first know what makes an ETL tool a good fit for a use case. The tool needs to be able to handle the case at hand while being future-proof to handle situations that might arise. This makes it necessary to understand the key aspects of what makes a tool suitable for a certain use case and what key aspects of a tool make it future-proof.

These key aspects were deduced partially from the performed literature study, but mostly from interviews conducted with developers from different teams from Topicus. The interviews were conducted in a semi-structured

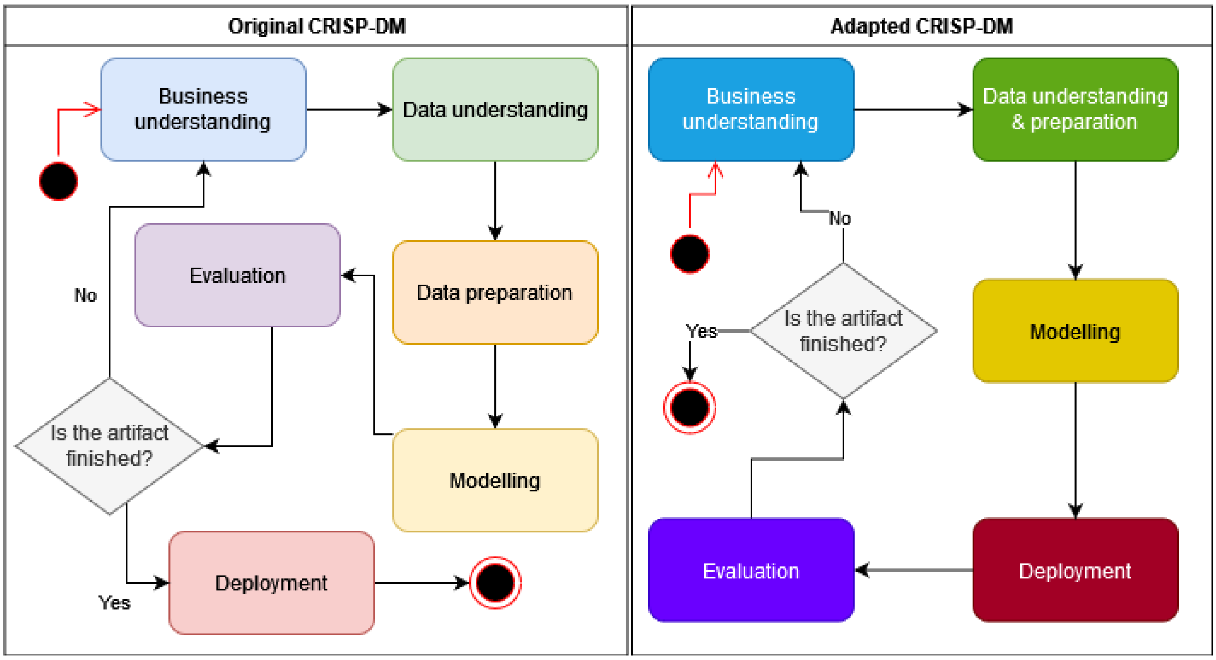


Figure 3.1: Comparison of the original CRISP-DM methodology and the adapted form

way. The questions can be found in appendix B.

The questions were designed to gather insights into their current ETL tool and its shortcomings, the developed ETL workflows, and their ideal situation. The questions were used as guidelines to gather all relevant information, however, if deemed necessary by the interviewer follow-up questions were asked to gain more information on certain topics or to clarify certain answers. The information gathered from these interviews is different from the requirements developers have for their use cases. For example, a developer might require assigning memory for each ETL pipeline. From this requirement, the aspect of resource control can be derived. The aspects gathered in these interviews are the topics of the requirements developers might have. Figure 3.2 displays this phase in blue. The results of the business understanding phase can be found in section 4.1.

3.3 Data understanding and preparation

The information gathered during the business understanding phase discussed in section 3.2 can now be used to continue with the data understanding and data preparation phases. In this phase, relevant data was collected and prepared [28, 46].

As mentioned in section 2, a list of currently available open-source ETL tools was created before this study [25]. These tools cover ETL tools in the broadest sense, as is later explained in more detail, the list of tools includes orchestrators, ETL tools, and data synchronization tools. Some offer cloud-based integration platforms as a service (iPaaS) as part of their application. Currently, iPaaS plays a big role in the shift from on-premise systems that move to the cloud. The reason for this broad definition of an ETL tool lies in the convergence of functionalities traditionally associated with iPaaS, which focuses on cloud integration, and standard ETL tools, typically used in on-premise systems [48]. This distinction has become increasingly ambiguous, as most tools now support connectivity to a wide range of data sources, as is shown later in this paper.

The next step is to see how these tools handle the key aspects found during the business understanding phase. This information was gathered in an aspect matrix documenting how each tool handles each key aspect. Next, this aspect matrix was converted into a scorecard, where each textual description of how a tool handles an aspect was converted into either a score indicating how well it can do this aspect or a simple true/false value of whether the tool has a specific aspect. This phase is displayed in green in figure 3.2. The results of the data understanding and preparation phase can be found in section 4.2.

3.4 Modeling

With the aspect matrix completed the modeling phase could begin. In the original CRISP-DM methodology this phase includes testing and assessing different machine learning models [28, 46] to develop the solution to the data mining problem. However, since in this paper, the CRISP-DM methodology is used as a design method, this phase was used to design and implement the recommendation system, which was done in the four steps listed below.

1. Create a questionnaire that developers must answer when looking for a new tool based on the aspect list deduced from the interviews

2. Convert the answers given to the questionnaire from human-readable text into numbers and boolean values

3. Create a logical model that would filter out incompatible tools for the given use case and rate the remaining tools on their capabilities that the user found important using the answers to the questionnaire created in the first step.

4. Create a front end for the user to view the results calculated by the logical model

These steps are also displayed in the yellow part of figure 3.2. The user can then use the results to do more targeted research and make a final decision. The final decision will still be left up to the user as the perfect tool might not exist and the compromises involved are highly subjective. The created questionnaire, the logical model with data conversion, and the front end can be found in section 4.3.

3.5 Deployment

Normally, the final phase of the CRISP-DM methodology is the Deployment phase. In this phase the designed model is made available to the end user [28, 46]. This can be as elementary as creating a dashboard or something more complex like creating a repeatable data mining workflow.

In the adapted CRISP-DM, the deployment consisted of two parts. The questionnaire was made available through Google Forms [24] as this is an easy way to create questionnaires and store the answers in an accessible way. The logical model and way to see the results were hosted on Streamlit [42], a free, open-source cloud hosting platform developed for users to create a data-driven app with simple Python code quickly. This phase is displayed in red in figure 3.2. The results of the deployment phase can be found in section 4.4.

3.6 Evaluation

The next phase is the evaluation phase. The traditional CRISP-DM focuses more on business requirements in this evaluation rather than technical performance as this would already be tested during the modeling phase [28, 46].

The adapted CRISP-DM evaluation phase consists of three parts which can be conducted in parallel. The first part was rating the compliance with the seven guidelines by Hevner et al. [27] as a form of self-evaluation of the process and the results. The second part was a survey to evaluate the usability and usefulness of the recommendation system. The last part was a case study, which showed if the logical model makes good suggestions. More detailed descriptions of each evaluation step can be found below. The different evaluations of this phase

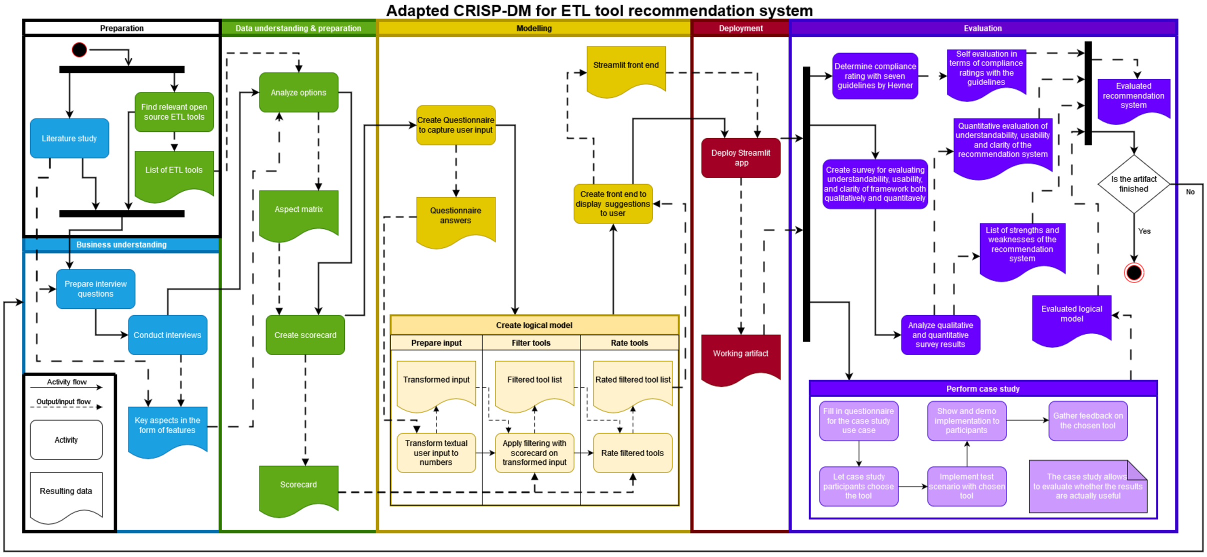


Figure 3.2: Methodology workflow

Guidelines	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation
Guideline 2: Problem relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Table 3.1: Brief description of the seven guidelines by Hevner et al. [27] used in the evaluation phase as seen in figure 3.2

are displayed in the purple section of figure 3.2. The results of each part of the evaluation can be found in section 5.

3.6.1 The seven guidelines

To evaluate if the process of designing the recommendation system was done properly and effectively, the seven guidelines by Hevner et al. were used. The seven guidelines by Hevner et al. were created in the context of Design Science (DS) in Information Systems Research as a way to ensure the quality of the DS research in information systems [27]. The use of these guidelines requires the researchers to critically self-reflect on the performed research and therefore help ensure the quality of the research. A brief overview of the guidelines can be found in table 3.1. In this paper, the artifact mentioned in these guidelines is the recommendation system.

While compliance with these guidelines offers a great way to ensure quality, not all guidelines are as important [45]. J. Venable published a study in which different quality insurance frameworks, among which the seven guidelines by Hevner et al., were evaluated on their importance and relevance on a scale of 0 - 10. The different frameworks were evaluated by editors of high-quality journals; program chair and committee members of the DESRIST conference (2006-2009); and authors of papers published at the DESRIST in 2006-2009. Compliance with all the seven guidelines together was not deemed as important as compliance with certain individual guidelines. The guidelines rated as most important by the participants were guidelines 1, 2, 3, and 4 each with ratings between 8.31-9.05. Guidelines 5 and 7 were deemed less important with ratings of 7.33 and 7.20 respectively while guideline 6 was the least important with a rating of 6.09.

This difference in ratings indicates that the created artifact and the relevance, evaluation, and novelty of said artifact (guidelines 1, 2, 3, and 4) are more important than what methods were used exactly to create the artifact (guidelines 5), how iterative the process was to complete the design (guidelines 6), and how the results are presented (guidelines 7). Therefore, the first four guidelines were taken as the basis to ensure the quality of the recommendation system, whereas the remaining guidelines are only briefly touched upon to see whether compliance was reached. Furthermore, Hevner et al. mention they advise against the mandatory use of their guidelines and instead recommend the researchers use their creative skills and judgment to determine when, where, and how to apply the guidelines. Therefore, we have determined a compliance rate with each guideline to the best of our ability as a way of self-reflection on the process of creating the recommendation system.

3.6.2 Survey

The next part of the evaluation is a survey. The questions of the survey can be found in appendix C. The goal of the survey is to evaluate the tool as a whole. The survey questions consisted of ratings from one to ten and open questions for respondents to elaborate on their ratings and gather suggestions for specific improvements. The results should give insights into how clear the suggestions were and how easy the recommendation system is to use. Respondents were asked to use the recommendation system multiple times with different scenarios in mind to see how it handles different use cases.

3.6.3 Case study

The last part of the evaluation was to see whether the results were useful. Therefore, a case study was set up in collaboration with Topicus .Finance. Topicus .Finance was looking to replace their current ETL tool with a new one to simplify their workflow. The case study consisted of three parts. First, the recommendation system was utilized to determine the optimal tool for their use case. Subsequently, a specific ETL process was replicated using the new tool. Lastly, the chosen tool could be evaluated with the ETL process running with the new tool. The most important factors for Topicus .Finance were ease of use, error logging, notifications, and scheduling.

Chapter 4

Design results

The following sections present the results of the design of the recommendation system. We discuss the results from the phases of the adapted CRISP-DM methodology in the same order as they are mentioned in chapter 3 except for the evaluation which is discussed in chapter 5.

4.1 Business understanding

The interviews, in conjunction with the trends identified in the existing literature, produced a list of key aspects and essential information to consider when evaluating a new ETL tool. The list of aspects and important information can be found in table 4.1. The description shows what the aspect or information entails or examples of what questions this knowledge will answer. In total four different teams, each consisting of two or three developers, were interviewed. Each team worked in a different division of Topicus where they worked on a different application which means the teams had different requirements for their ETL process and each team had a different use case. This is critical to ensure the derived key aspects are generalizable across use cases.

Aspects such as schema changes, and loading of data were first derived from the literature, but these results were corroborated in the interviews by the majority of the developers as important. Other aspects such as Monitoring, scheduling, and Documentation were only highlighted by developers as important.

4.2 Data understanding and preparation

This section discusses how different parts of the scorecard are set up and clarifies any discrepancies in the aspects presented in the scorecard. This section is divided into two parts, the filtering aspects and the ranking aspects. These topics are further discussed during the logical model presented in the modeling phase in chapter 4.3.

The list of aspects presented in section 4.1 was slightly extended with the extracting being split up into extraction from a database (DB), extraction from a file, extraction from an API, and extraction from other applications to cover different situations separately rather than as a whole. Similarly, the loading was split up into loading to a DW, loading to a data lake (DL), loading to a lakehouse (LH), and loading to a different application. As mentioned in section 3.3, we first created an aspect matrix for the different tools to see how they handle different aspects. These results consist of descriptions for each tool for each aspect. These results can be found in the following spreadsheet [1].

Based on these descriptions, a scorecard was created which converted this text into a score indicating how well, if at all, a tool can handle an aspect. The scorecard is shown in table 4.2. Since the table is very long, the table is split up into multiple parts separated by a white line after which new column names

[1] Full link to spreadsheet: https://docs.google.com/spreadsheets/d/14DavziMyOq5kswY-1HteA8oD3N6Qz9QBHkj8f0pha8Q/edit?usp=sharing

for the next tools are written.

4.2.1 Filtering aspects

The aspects *ETL tool* up until and including *Encryption* are given values of 0, 0.5, or 1 and are used to narrow done the possible recommendations. A 0 indicates that this tool is incapable of doing this or unsuitable for this task. For example, Airbyte is unsuitable as an orchestrator and does not have event triggers. A 1 means this tool is capable of this aspect or is suitable for this task. For example, Airbyte is meant as a data synchronization tool and it supports cloud hosting and even offers a cloud integration platform. Lastly, tools that are capable of doing a task but are not designed for this purpose or require some user-created logic receive a 0.5. For example, Apache Beam is not designed as a data synchronization tool but can be used as one. The 0.5 will ensure a tool is considered but will generally score lower than a tool specifically designed for the same purpose.

The first four rows, *ETL tool*, *Orchestrator*, *Data sync tool*, *DW tool*, indicate what the tool was designed for. An ETL tool is defined as a tool where data moves through it. An orchestrator is a tool that, as the name suggests, orchestrates the workflow. These kinds of tools can call other software to perform tasks and streamline an ETL pipeline. A data sync tool is a tool that only transfers data from a source to a destination. These tools are effective for ELT where transformations are done after the data is loaded into the destination. Lastly, DW tools can extract data from different sources but act as the destination themselves. These tools have integrated storage and can be used directly to build dashboards and reports.

The row *Add on tool* indicates if the tool can be used alongside other tools. For example, Apache Spark integrates well with Apache Hadoop and Apache Hive and can therefore be an add-on to either. Another example is DBT, which is a tool designed only for transformations. Models created in DBT can be used in almost all considered tools. DBT is a special case for these first five rows, as it received a 1 in all of them. This is not because DBT is this outstanding tool capable of all, but rather since it is specialized only in transformation, it should be taken into account for every use case as an add-on tool.

The row labeled *CDC* indicates if a tool can capture only changed data as mentioned in table 4.1 for the *Change data capture* aspect. A few tools are capable of change data capture themselves, the other tools all got a 0.5 as it is always possible for the user to implement this themselves or the tool integrates with Debezium [13], an open-source distributed platform for change data capture.

The five rows following, *Docker hosting*, *Application hosting*, *Library hosting*, *Cloud hosting*, and *Own cloud*, are all related to the *hosting* aspect described in table 4.1. Some tools are only hosted as docker containers, or as stand-alone applications, while other tools can be hosted in multiple ways. Some tools even offer a cloud service for hosting all the user's ETL pipelines in a cloud environment optimized for this tool.

Next, *Code*, *Scripting*, *Config files*, and *No-code* relate to the implementation of ETL pipelines. Also see *Code or low-code* in table 4.1. The *Code* aspect indicates an application uses pure programming to implement an ETL pipeline. *Scripting* is more low-code, where the pipeline is mostly implemented with no-code building blocks that can be configured but there are several options for using scripting to perform certain transformations or tasks. *Config files* indicate using configuration files to implement the entire ETL pipeline or to set certain properties. These files are usually XML, JSON, or YAML files. Lastly, *No-code* indicates there are no options for programming or configuration files, there is only a User Interface in which the pipelines can be designed and configured.

The following four aspects, *Integrated scheduling*, *CRON*, *Event triggers*, and *Workflow triggers*, are all related to the *scheduling* aspect from table 4.1. The first, *Integrated scheduling* indicates if a tool has its own scheduling capabilities or if it requires another tool like an orchestrator. The other three, indicate if the tool supports that type of scheduling or trigger. The last row that uses the 0, 0.5, or 1 system is *Encryption*. This row indicates if a tool supports encrypting data or masking sensitive data. This row is related to the *Security* aspect from table 4.1.

4.2.2 Ranking aspects

The rows *Resource control* up until and including the last row *Training*, are meant to ensure a higher ranking for tools that better handle aspects the user indicates as important. These rows were scored based on how capable a tool is for this specific task. This score was determined by first categorizing all tools for each aspect. Depending on how many categories were defined the best tools would get a score equal to the number of categories while the worst tool would get a 1. The number of categories was determined by how many distinct factors played a role in the aspect.

For example, *Resource control* was given a score of 1 through 4, where 1 means no control or information was available on resource control; 2 means the users could review the resources that were used afterward; 3 means the user can set a maximum amount of resource that a workflow is allowed to use; and 4 means full control over resources. However, a task as *Training* was given a score of 1 through 3, where 1 means there is basic documentation but it might be cluttered or the examples might be confusing; 2 means the documentation and examples are clear; and 3 means the documentation and the examples were coherent and extensive and there was something extra to enhance the learning experience, for example, a demo environment or video tutorials. In this last case, only three categories were necessary to divide the tools.

The row *Programming languages* shows all programming languages or file types that can be used for coding, and scripting or configuration files respectively. The remaining rows are related to the similarly named aspects described in table 4.1 and are therefore self-explanatory. The exceptions are the source and destination types which all relate to *Extracting* and *Loading* respectively. Furthermore, the *Training* row is related to the *Documentation* aspect but was changed to training to embody the onboarding of the new tool entirely rather than just the documentation's quality. The *Source* row indicates how capable the tool is at handling many different sources. The type of sources and destination that follow are indicators of how well the tools can work with this type of source or destination.

4.3 Modelling

This section is divided into three parts. First, the questionnaire users have to fill out is shown. Second, the logical model created to generate the suggestions is presented. Lastly, the Streamlit front end is shown. All three parts combined show the creation of the recommendation system that helps users pick a new ETL tool. From this point forward, the recommendation system is referred to as the ETL picker. The ETL picker can be viewed and used through this link [2]

4.3.1 Questionnaire

The questionnaire was designed to gather information on the user's requirements. This entails the requirements on the key aspects previously defined. To match these requirements to the tools the questions are related to the same topics and aspects as the aspect matrix and scorecard shown in section 4.2. The questions were designed to allow for

[2] Full link: https://forms.gle/d4qSudMVfref8fLA6

different specific scenarios as well as broad exploratory cases where not everything is set in stone yet and the user mostly wants to find out what tools are available based on some principles they do already have in mind.

Furthermore, the topics do not appear literally as they are presented in table 4.1 or 4.2, rather the questions require the user to critically think about their use-case and requirements rather than directly asking them if they want a certain aspect. This ensures the user does not simply want all the aspects even if these are not necessary. This also, to a certain extent, ensures a tool is available for their use case. As is discussed in more detail in section 4.3.2 no tool may cover everything for the use case of the user.

The questionnaire itself was hosted as a Google form. This format was chosen for two reasons. The first was that it is easy to set up and maintain. Creating a Google form is a straightforward process while allowing for the required degree of complexity that this questionnaire brought. The form supports the required answer types, such as checkboxes and multiple-choice. The second reason was that answers were stored in a Google sheet. This made it convenient to retrieve the answers from a certain person to calculate and show their results.

The questionnaire is divided into the following six categories.

1. General & storage related questions
2. Data
3. Technical architecture & security
4. Implementation
5. Monitoring & scheduling
6. Version control, community & learning

The questionnaire starts with a brief explanation of what the ETL picker is and how it works, followed by questions regarding each of the six categories. An overview of the questions and the kind of answer that is expected of the user can be found in table 4.3. Furthermore, it contains further explanations about the options the user can choose from if applicable. The complete questionnaire is displayed in appendix D.

4.3.2 Logical model

After users fill in the questionnaire part of the ETL picker, the logical model that was created will calculate a score for all tools that fulfill the requirements. This logical model consists of three parts, a preparation part, a filtering part, and a rating part. The preparation part gathers the answers from the user and transforms them into usable data. This mostly means transforming text fields into the names of the rows of the scorecard such that they can be immediately used during filtering and rating. For example, if a user checks the boxes for *ETL tool* and *Data synchronization tool*, these will be transformed to *ETL tool* and *Data sync tool* to match the rows of the scorecard table (4.2). Other answers, such as the question on transformations, are transformed into a numerical value based on the answer which indicates how important this is to the user. This will ensure that during the rating part, tools that score high on important aspects will rank higher than tools that score high on unimportant aspects.

Filtering

Each tool will start with a score of 1. The filtering part will apply the rows *ETL tool* up to and including *Encryption* from the scorecard (see table 4.2) to the scores of the tools based on the answers given by the user in the questionnaire, resulting in any tools that do not comply with the use case to be dropped. There are a few interesting parts to note.

First, if a user only selected *complete data warehouse tool including storage* the model will immediately stop filtering and move on to rating as only three tools fall under that category. Suppose a user specifically did not check this type of tool but did indicate they do or might want integrated storage. In that case, the DW tools are still included in the filtering even though the user did not check this type of tool.

Second, questions where multiple options can be selected, such as question one about the type of tool, the question about hosting, and the question about implementation, result in applying each applicable row of the scorecard. For example, if the user indicates they would like a tool that can be hosted in docker or as a stand-alone application, both the rows *Docker hosting* and *Application hosting* from the scorecard (4.2) will be applied. In this example any tool that is hosted either in docker or as a stand-alone application will be considered.

Third, the rows *Integrated scheduling*, *Own cloud*, and *Encryption* from the scorecard (table 4.2) are only applied if the user indicated they do not want a separate tool for scheduling, they are interested in a cloud environment offered by the tool, and they indicated they are dealing with sensitive data respectively.

Lastly, change data capture (CDC) is the only aspect during filtering that does not necessarily result in a zero or one score. This is because the need for CDC is determined based on two questions. The first question is if the data is too large to be dropped and loaded every time, which is used to see if CDC is necessary in the first place, which does result in a zero or one. The second question is how often the data needs to be loaded in, if data is only loaded in less than once a day, the need for CDC is less important than if it has to be near real-time. Based on the answer to this question, this zero or one is multiplied by a number from 1-5 resulting in a score from 0-5. While this is also already rated based on how well the tools can do CDC, if the user requires CDC, tools that are incapable of CDC will be filtered out.

Ranking

The last step is to rate the remaining tools based on the remaining rows from the scorecard combined with the remaining answers. For the majority of these rows, the rating score was calculated by increasing the current rating score of a tool by the value of the scorecard multiplied by the value of the answer. Since most of the remaining answers were converted into a number and the values from the scorecard are already numeric, these can be multiplied and added up easily for all aspects of each tool.

After adding the score of an aspect to the current rating score, the scores were normalized using a min-max normalizer. This ensures that all scores are always between 0 and 1 which in turn safeguards an equal contribution of all aspects to the final rating score. Suppose scores are not normalized after each step. In that case, aspects that were divided into more categories, and can therefore receive a higher score, such as *Resource control* which can receive a score as high as 4, would be seen as more important than an aspect such as *Monitoring* which can only receive a maximum score of 3. The goal of the ETL picker is to let the user determine which aspects are important with their answers to the questionnaire.

There are a few exceptions where it was not directly possible to add the score in this way. The row labeled *Programming languages* was divided into programming language and configuration file types. The right set of answers was chosen depending on whether the user wanted to use programming/scripting languages or configuration files. The score reflected these preferences by adding one point to the tool for each of the programming/scripting languages or file types the tool supports. Furthermore, not all rows for the source types (*DB source, File source, API source*, and *Application source*) and destination types (*DW destination, DL destination, LH destination* and *Application destination*) were applied.

Only the rows that correspond to the answers to the question regarding source types and storage destinations respectively were applied.

4.3.3 Streamlit front end

As mentioned, the suggested tools will be displayed to the user through Streamlit. The results can be presented to the user by creating a straightforward front end. The user is asked to fill in the email address they filled in on the questionnaire, as can be seen in figure 4.1a.

After the user enters their email address and presses the button labeled *'See results'* all answers from the Google Sheet matching that email address will be fetched and the tools will be filtered and ranked. This means a user can fill in the ETL picker multiple times for a different type of tool and see all their results in one go for each scenario. If the user enters an email address that is not found in the answers or enters an invalid email address, the user will be shown a message that no results are found for that email address or be asked to enter a valid one.

The first thing the user will see is a small text briefly explaining the scores the user is about to see. As mentioned, the ratings will be given as a score from 0 to 1. Where a score of 0 is the least compatible, however, it should still be capable of handling the use case described by the user. After this introductory text, the date and time of when the questionnaire was filled in are shown such that the user can distinguish the different times they filled in the questionnaire. After the date and time, the results are shown in a small matrix with the name of the tool and its final score ordered from highest to lowest score.

Lastly, any useful info about their results is shown. There are four info messages, one message for if DBT is in the results, two messages for when the user indicated they wanted integrated storage either for sure or maybe, and one last message for if the user indicated scheduling can be done with a different tool. An example of a few of these messages is shown in figure 4.1b.

If no tools fit the user's answers, a message will be displayed stating their requirements are too specific and can not be matched to any tools. They are suggested to change either the implementation method or the hosting options as these two aspects have the greatest impact during filtering.

4.4 Deployment

As mentioned the deployment was done with Streamlit. The application itself was shown in section 4.3.3. Deploying an app with Streamlit only requires your code to be saved in a GitHub repository. It is also possible to start a Streamlit app from a provided template after which it automatically creates a GitHub repository. After connecting and telling the Streamlit back end which file to run, Streamlit takes care of the rest. While deploying an app it is possible to incorporate authorization details in a secrets file which can be used in the app without anyone seeing the actual content. Furthermore, after deploying the developer gets info logging in case an error occurs within your app. The deployment process with Streamlit was easy and generalizable. For more detail on how to deploy with Streamlit, see appendix E

Begin of table	
Aspect	**Description**
Product version	Document the version that is taken into consideration. Furthermore, it speaks to how evolved the tool is, if it is still very novel that might be a reason for people not to choose it
Hosting	How is the tool hosted? Can it be hosted in a docker container? Is it a stand-alone application? Is it a programming library
Resource configuration	How much control does the user have over resources that an ETL job/workflow/pipeline can use?
Extracting	Extracting data might have to be done from multiple sources of different types, like a database and several files stored in cloud storage, or an API. It is important to know how a tool handles these different situations.
Transformations	Data might have to be transformed during the ETL process, this can be simple filtering or more advanced customized transformations that are developed by the user. Are these features available out of the box or does the tool integrate with another tool for transformations?
Loading	Like extracting, the data should also be loaded into a destination. For example, a data warehouse, data lake, or lakehouse.
Code or low-code	How is the ETL process implemented? Some users prefer pure programming code, while others prefer a more low-code/no-code UI. Furthermore, this entails what programming languages the tool supports either for scripting or full programming language for workflow/pipeline implementation as well as what type of configuration files a tool might use if they have any.
ETL vs ELT	Is the tool more geared towards ETL or ELT? What purpose was this tool designed for? Is it more designed for raw data synchronization between storage or does data move through the tool?
Orchestration	How suitable is the tool to orchestrate all the user's ETL jobs/workflows/pipelines? Is data meant to move through this tool or is it meant to coordinate a workflow? Very much correlated with *ETL vs ELT*
Change data capture	Can the tool capture new data inserted since it was last fetched from the sources or does the user have to filter this in their transformations?
Schema changes	If the source or destination data types are changed or a column is added or removed, how does a tool handle this? Can it do this automatically or does the user have to do this?
Monitoring	How can the user monitor the jobs/workflows/pipelines that have run and the jobs that are scheduled to run? What kind of error logging is there?

Continuation of Table 4.1	
Aspect	**Description**
Triggers/scheduling	How can jobs/workflows/pipelines be scheduled or triggered? Can the tool do this by itself or does it require an orchestrator?
Security	What security options does the tool offer the user? Are there options for data encryption?
Versioning	Does the tool integrate with version control platforms like Git? How are jobs/workflows/pipelines stored and is this suitable for version control and a review process?
Documentation	How clear is the documentation? Are there tutorials? How clear are the examples that are given?
Community	How many stars and contributors do their Github page have? How active is the community in helping each other with problems?
End of Table	

Table 4.1: List of aspects and important information to take into consideration when choosing a new ETL tool based on interviews and trends found in the literature with a brief description of what they entail

Begin of table

Aspect	Airbyte	Apache Airflow	Apache Beam	Apache Camel	Apache Druid	Apache Hadoop	Apache Hive
ETL tool	0	0	1	1	0	0	0
Orchestrator	0	1	0	0.5	0	0	0
Data sync tool	1	0	0.5	0.5	0.5	0	0
DW tool	0	0	0	0	1	1	1
Add on tool	0	0	0	0	0	0	0
CDC	1	0.5	0.5	0.5	0.5	0.5	0.5
Docker hosting	1	1	1	1	1	1	1
Application hosting	0	0	0	0	0	0	0
Library hosting	0	0	1	1	0	0	0
Cloud hosting	1	1	0	0	0	0	0
Own cloud	1	0	0	0	0	0	0
Code	0	1	1	1	1	1	1
Scripting	0	0	0	0	0	0	0
Config files	0	0	0	0	0	0	0
No-code	1	0	0	0	0	0	0
Integrated scheduling	1	1	0	1	1	0	0
CRON	1	1	0	1	1	0	0
Event triggers	0	0	0	1	0	0	0

Continuation of Table 4.2

Aspect	**Airbyte**	**Apache Airflow**	**Apache Beam**	**Apache Camel**	**Apache Druid**	**Apache Hadoop**	**Apache Hive**
Workflow triggers	0	1	0	1	0	0	0
Encryption	1	1	1	1	1	1	1
Programming languages	-	Python, SQL	Java, Python, Go, SQL, YAML	Java, SQL, XML, YAML, Groovy, Kotlin	SQL	Python, Java, C++, C#	SQL
Resource control	4	2	3	3	2	2	2
Monitoring	3	3	1	1	1	3	1
Sources	3	2	3	3	3	1	1
DB source	3	3	3	3	3	1	1
File source	3	3	3	3	3	1	1
API source	3	3	3	3	3	1	1
Application source	3	2	2	2	2	1	1
Transformations	1	3	3	3	3	2	3
Schema changes	4	1	3	3	2	1	2
DW destination	3	3	3	3	3	1	3
DL destination	3	3	3	3	2	3	1
LH destination	3	3	2	2	2	2	1
Application destination	3	2	2	2	1	1	1
Security	3	3	1	3	3	2	2
Version control	1	3	3	3	2	2	2
Community	4	5	3	3	4	4	3
Training	3	3	3	2	3	1	2

Aspect	**Apache Hop**	**Apache Kafka**	**Apache Nifi**	**Apache Seatunnel**	**Apache Spark**	**Cloud Query**	**Dagster**
ETL tool	1	0	1	0	0	0	0
Orchestrator	0.5	0	0	0	0	0	1
Data sync tool	0.5	1	0.5	1	0	1	0
DW tool	0	0	0	0	1	0	0
Add on tool	0	0	0	0	1	1	0
CDC	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Continuation of Table 4.2							
Aspect	**Apache Hop**	**Apache Kafka**	**Apache Nifi**	**Apache Seatunnel**	**Apache Spark**	**Cloud Query**	**Dagster**
Docker hosting	1	1	0	1	1	1	1
Application hosting	1	0	1	0	0	0	0
Library hosting	0	0	0	0	0.5	0	1
Cloud hosting	0	0	0	0	0	1	1
Own cloud	0	0	0	0	0	0	1
Code	0	1	0	0	1	0	1
Scripting	1	0	1	0	0	0	0
Config files	0	0	0	1	0	1	0
No-code	1	0	1	1	0	0	0
Integrated scheduling	1	1	1	1	0	0	1
CRON	1	0	0	0	0	0	1
Event triggers	1	1	1	1	0	0	1
Workflow triggers	1	0	0	0	0	0	1
Encryption	1	1	1	0	1	0	1
Programming languages	SQL, Shell, Python, Javascript, Groovy	Java, Scala, SQL	Jython, Groovy, Javascript, JRuby, Clojure, SQL	JSON, HOCON	Python, SQL, Java, Scala, R	YAML	Python, SQL
Resource control	4	4	4	3	3	3	3
Monitoring	1	1	3	3	3	2	3
Sources	3	2	3	3	2	3	3
DB source	3	1	3	3	3	3	3
File source	3	3	3	2	3	3	3
API source	3	3	3	3	3	3	3
Application source	2	2	2	3	2	3	3
Transformations	3	1	2	2	3	1	3
Schema changes	1	1	3	2	4	4	2
DW destination	3	2	3	2	3	3	3
DL destination	3	2	3	2	3	2	3
LH destination	2	2	2	2	2	1	3

Continuation of Table 4.2

Aspect	Apache Hop	Apache Kafka	Apache Nifi	Apache Seatun-nel	Apache Spark	Cloud Query	Dagster
Application destination	2	2	2	3	2	3	3
Security	3	3	3	3	3	2	1
Version control	2	3	2	3	3	3	3
Community	1	4	3	3	5	3	4
Training	3	1	1	3	3	3	3

Aspect	DBT	Kestra	Knime	Mage	Meltano	Pentaho	Prefect
ETL tool	1	0	1	1	0	1	1
Orchestrator	1	1	0.5	1	0	0.5	1
Data sync tool	1	0	0.5	1	1	0.5	1
DW tool	1	0	0	0	0	0	0
Add on tool	1	0	0	0	1	0	0
CDC	0.5	0.5	0.5	0.5	1	0.5	0.5
Docker hosting	1	1	0	1	1	1	1
Application hosting	0	0	1	0	0	1	0
Library hosting	0	0	0	1	0	0	1
Cloud hosting	1	1	1	0	0	0	1
Own cloud	1	1	1	0	0	0	1
Code	1	0	0	1	0	0	1
Scripting	0	1	1	0	0	1	0
Config files	0	1	0	0	1	0	0
No-code	0	0	1	0	0	1	0
Integrated scheduling	0	1	1	1	1	1	1
CRON	0	1	1	1	1	1	1
Event triggers	0	0	0	1	0	1	1
Workflow triggers	0	1	1	1	0	1	1
Encryption	1	1	0	1	0	1	1
Programming languages	SQL	YAML	Python, R, Javascript, SQL	Python, SQL	YAML	SQL, Python, R	Python, SQL
Resource control	2	3	3	2	2	2	4
Monitoring	2	3	3	3	1	2	3
Sources	1	3	3	2	3	3	2

Continuation of Table 4.2							
Aspect	**DBT**	**Kestra**	**Knime**	**Mage**	**Meltano**	**Pentaho**	**Prefect**
DB source	2	3	3	3	3	3	3
File source	1	3	2	3	3	3	3
API source	1	3	3	3	3	3	3
Application source	1	3	2	2	3	2	2
Transformations	3	3	2	3	2	3	3
Schema changes	4	1	1	1	1	1	2
DW destination	2	3	3	3	3	3	3
DL destination	1	3	2	3	3	3	3
LH destination	1	3	3	3	3	2	3
Application destination	1	3	2	2	3	2	2
Security	1	3	1	1	1	3	2
Version control	3	3	1	3	3	2	3
Community	3	3	2	3	2	3	4
Training	1	3	2	3	3	1	3

Aspect	**Luigi**	**Petl**	**PyGram ETL**	**R_etl**	**Singer**		
ETL tool	0	1	1	1	0.5		
Orchestrator	1	0	0	0	0		
Data sync tool	0	0.5	0.5	1	1		
DW tool	0	0	0	0	0		
Add on tool	0	1	1	0	1		
CDC	0.5	0.5	0.5	0.5	0.5		
Docker hosting	0	0	0	0	0		
Application hosting	0	0	0	0	0		
Library hosting	1	1	1	1	1		
Cloud hosting	0	0	0	0	0		
Own cloud	0	0	0	0	0		
Code	1	1	1	1	1		
Scripting	0	0	0	0	0		
Config files	0	0	0	0	1		
No-code	0	0	0	0	0		
Integrated scheduling	1	0	0	0	0		
CRON	1	0	0	0	0		

Continuation of Table 4.2							
Aspect	**Luigi**	**Petl**	**PyGram ETL**	**R_etl**	**Singer**		
Event triggers	0	0	0	0	0		
Workflow triggers	1	0	0	0	0		
Encryption	1	1	1	0	1		
Programming languages	Python, SQL	Python, SQL	Python, SQL	R, SQL	Python, JSON, SQL		
Resource control	2	1	1	1	2		
Monitoring	3	1	1	1	1		
Sources	2	2	2	2	3		
DB source	2	3	3	2	3		
File source	2	3	3	2	3		
API source	2	3	3	2	3		
Application source	2	2	2	1	3		
Transformations	3	3	3	3	2		
Schema changes	1	1	1	1	3		
DW destination	1	3	3	2	3		
DL destination	1	2	1	1	3		
LH destination	1	2	1	1	3		
Application destination	1	2	2	1	3		
Security	1	1	1	1	1		
Version control	3	3	3	3	3		
Community	4	2	1	1	1		
Training	3	3	2	1	1		
End of Table							

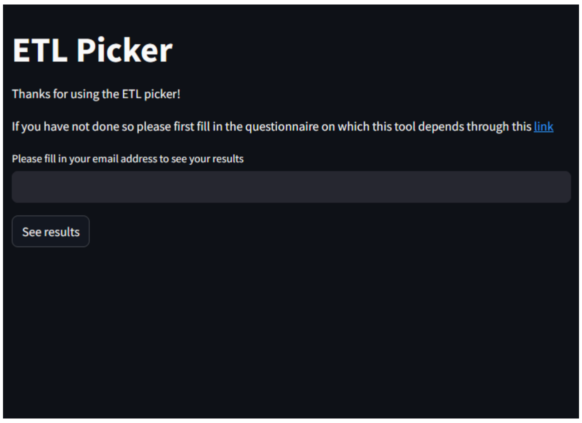
Table 4.2: The scorecard created based on the aspect matrix

Begin of table		
Category	**Question**	**Kind of answer**
General & storage	Are you looking for an ETL tool, orchestrator, data synchronization tool, or complete data warehouse including storage? Pick any that might apply	Checkboxes for each option, user can check any that they want to include
	Do you already have a storage destination?	Multiple choice, depending on the answer tools including storage will or will not be taken into account

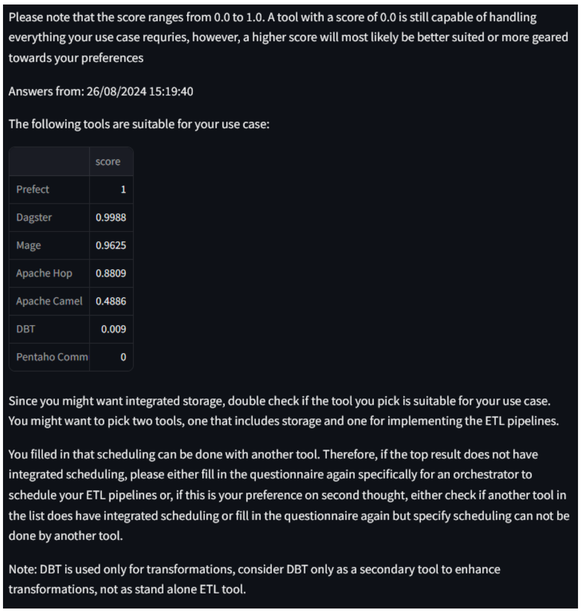
Continuation of Table 4.3		
Category	**Question**	**Kind of answer**
Data	Do you need to combine data from many different (types of) sources?	Multiple choice
	What type of sources do you have?	Checkboxes for Databases, files, APIs, and other applications
	How much does this data need to be transformed in order to fit your needs?	Multiple choice
	How often does the source or destination schema change?	Multiple choice
	How will your data be stored?	Multiple choice, includes structured, unstructured, both, or in another application.
	How often does new data need to be loaded in?	Multiple choice, including near real-time, every hour, every half day, every day, or less than once a day.
	Is the data size too large to drop and refill the entire table every time?	Yes/no, indicates if Change Data Capture is necessary
Technical architecture & security	How would you like to host the application?	Checkboxes for Docker, stand-alone application, Programming library, and cloud hosting. Multiple can be selected.
	If you are considering cloud hosting, what kind of cloud provider would you like to use for running your ETL processes? Please leave blank if you are not considering cloud hosting.	Multiple choice, used to see if the application with their own cloud hosting options should be suggested.
	What minimum resource configuration requirements do you have?	Multiple choice, four options ranging from full control to no requirements
	If resource configuration is done through config files, what types of configuration files would you like to use?	Checkboxes for HOCON, JSON, XML, YAML
	Do you already have security in place for hosting and running your ETL securely or do you want a tool to help you with that?	Multiple choice, options are available if security is already in place and if security will be taken care of outside of the tool
	Are you working with a lot of sensitive data that needs to be masked or encrypted?	Yes/no, indicates if encryption and/or masking options should be available.
Implementation	How do you prefer to implement your ETL pipelines?	Checkboxes for only code, no-code with scripting, configuration files, and pure no-code blocks. Users can select all that they prefer.
	If you want to use programming or scripting, what programming language(s) do you want to code in? Leave empty if not applicable	Checkboxes for each programming language found during the data understanding and preparation phase.

Continuation of Table 4.3		
Category	**Question**	**Kind of answer**
Monitoring & scheduling	How important is monitoring for your use-case?	Rating from 1-5
	How extensive monitoring is required?	Rating from 1-5
	What kind of scheduling do you want?	Checkboxes for CRON/time-based scheduling, event triggers, and workflow triggers
	Can scheduling be done with another tool?	Yes/no, indicates if the tool should have its own scheduling or if a separate orchestrator or scheduler can be used.
Version control, community & learning	How important is version control?	Rating from 1-5
	How important is a strong community?	Rating from 1-5
	How important is training and onboarding of the new tool? This includes documentation, (video) tutorials, and other guidelines	Rating from 1-5

Table 4.3: List of questions and answer types of the questionnaire



(a) Streamlit results front end where the user can enter their email address



(b) ETL picker example results page that is shown to the user

Figure 4.1: Streamlit front end

Chapter 5

Evaluation results

This chapter discusses the result of the evaluation phase of the adapted CRISP-DM methodology. As mentioned in section 3.6, the evaluation consists of three parts. We first discuss the compliance with each of the guidelines by Hevner et al. [27]. Next, we look at the results of the evaluation survey and lastly, we look at the case study. Any improvements regarding the recommendation system that were mentioned or found during the evaluation is further discussed in section 6.4

Guidelines	Rating
Guideline 1: Design as an Artifact	Above average
Guideline 2: Problem relevance	Average
Guideline 3: Design Evaluation	Great
Guideline 4: Research Contributions	Average
Guideline 5: Research Rigor	Average
Guideline 6: Design as a Search Process	Below average
Guideline 7: Communication of Research	Below average

Table 5.1: The seven guidelines by Hevner et al. [27]

5.1 The seven guidelines

The compliance was rated by the researchers as *poor, below average, average, above average* or *great*. An overview of the ratings can be found in table 5.1. As mentioned before in chapter 3.6.1 the first four guidelines are more important than the last three [45]. Therefore, the focus is on those first four.

Compliance with the first guideline is rated as above average. The developed ETL picker is a viable artifact that people can use to find a new ETL tool. Therefore, in and of itself the ETL picker is compliant with this first guideline. In this case, compliance with this guideline is the easiest of them all as the goal of the research was to produce a working artifact to help users find a new ETL tool. Since the artifact does work compliance is immediately achieved. Whether the ETL picker has helpful suggestions is not yet important for this guideline as this is covered in other guidelines and evaluations as part of this study.

Compliance with the second guideline is rated as average. The ETL picker does address a problem that is relevant to certain people. However, the extent of this relevance is difficult to determine. The ETL picker was designed mostly by working closely with Topicus developers. Although developers from different teams that all faced the problem of finding a suitable ETL tool were used in gathering information, it is not guaranteed that this is considered a problem more widely. The results from the survey discussed in section 5.2 should give more insights into this as well, however, that part of the evaluation focuses more on the usability of the ETL picker. The average compliance rate was given to this guideline with the assumption that if multiple teams in a company as large as Topicus came across this problem on their own, the results of this study at least solved the problem for these people. A higher compliance rate would have been achieved if the problem was also identified outside of Topicus.

Compliance with the third guideline was rated as great. According to the researchers, the evaluation of the ETL picker is considered extensive enough to conclude the usability and quality of the developed artifact and is therefore given a compliance rating of great with the third guideline. By applying the guidelines created by Hevner et al. [27] the process of developing the artifact as well as the artifact itself is evaluated on different factors. Even though the researchers themselves rate the compliance, it allows for self-reflection on the process and helps identify limitations in the conducted research. A survey was conducted to evaluate the usability and quality of the artifact and limitations, suggestions, and other improvement points could be gathered for further development. Lastly, the case study helps determine whether the suggestions are helpful and whether the logical model performs well.

Compliance with the fourth guideline was rated as average again. The idea behind the ETL picker is not new in and of itself. The research contribution lies in the methodology employed to develop the artifact and in the artifact itself. The research done before the start of the development produced useful insights into current themes and trends in current research that is performed in the domain of Data warehousing which was used as the basis for both the interviews and the aspect matrix during the development of the ETL picker. Furthermore, this previous study also found a list of tools that are considered as suggestions for the ETL picker. Moreover, the employed methodology is a repeatable process that can be easily adapted to fit similar problems, which is an even greater contribution.

The compliance with the fifth, sixth, and seventh guidelines were rated as average, below average, and below average respectively. The research rigor (guideline 5) was rated as average as the methods used to obtain the previous results and the results presented in this paper have considerable scientific substantiation and the results themselves have significant implications.

Compliance with the design as a search process (guideline 6), which was deemed the least important guideline [45], was rated as below average due to the limit in the cyclic approach. Although this is not seen as the most important guideline for developing a quality artifact, the cyclic or iterative approach has been around for a long time for good reason. The CRISP-DM methodology also should be used as an iterative process where problems are derived in the evaluation phase and solved in a new iteration [28, 46]. While this study conducts an evaluation that gives rise to some problems, which is highlighted and discussed in sections 5.2 and 6.4, there is only one iteration. The suggested improvements are not yet implemented afterward and are not re-evaluated with the same participants.

The last guideline received a compliance rate of below average as well. While the design process and results are properly presented in this paper for scientific use, communication to the intended users of the ETL picker, both technologically oriented and management-oriented, can be improved. As shown in the survey results presented in section 5.2 there are some misconceptions about the workings and suggestions produced by the ETL picker. Therefore, the communication to users can be improved to ensure their expectations are kept realistic.

5.2 Survey

The survey results can be divided into two parts, quantitative and qualitative results. The quantitative results encapsulate all questions that asked the respondent to give a rating. The qualitative results encapsulate the other questions the respondents could answer freely.

Aspect	Average	Standard deviation
Understandability	8	0.894
Usability	7.167	0.983
Question clarity	7.833	1.329
Result clarity	5.667	2.733
Overall score	6.833	1.722

Table 5.2: Average and Standard Deviation of each survey question

5.2.1 Quantitative results

The results of the quantitative part of the evaluation are shown in table 5.2. The most interesting results are the *result clarity* score, these are the lowest of all but do have the highest standard deviation. The minimum result this question received was a 2, whereas the maximum was a 10. This was also reflected in the qualitative results as most comments were left regarding the results and how to improve them.

5.2.2 Qualitative results

For each quantitative feature, an open question was added for elaboration. We go over several interesting comments that the participants left. First, even though the understandability of the ETL picker was rated highly, several comments were given that it does require knowledge of the domain and the language used. Which was later corroborated in the comments on usability and questions clarity. Suggestions were made to add a definitions list at the start such that everybody is on the same page.

Second, comments on usability mostly included the looks and the editing of their response to see how this affects their suggestions. As is discussed in section 6.4, this can be done as an improvement by incorporating the questionnaire within Streamlit.

Third, the participants missed questions regarding pricing; error handling and retry policies; and integration with other tools. Pricing was not added as all tools are free to use. Several tools do offer a paid version or a paid cloud environment. A question was added regarding this cloud environment, however the focus of that question was not so much on the cost aspect. The error handling and retry policy were taken into consideration at first, but it was decided to combine them with monitoring aspects. Perhaps this should have been made more clear. The last suggestion on integration with other tools was also corroborated in the final question where the participant could leave final comments and suggestions. This is a topic further discussed in the future works section on connectedness of tools 7.4.1.

Lastly, there were comments on the reasoning behind the results. Participants would like to see why some tools would do better than others and which of their requirements are met.

5.3 Case study

Currently, Topicus .Finance uses a tool called Pentaho Community Edition. Their experience with this tool has become deterred over the years and therefore they are looking to replace it. Their main concerns regarded ease of use, error logging and notifications, and scheduling.

Topicus .Finance filled in the ETL picker and received the results as shown in table 5.3. As can be seen, Prefect is the most suitable tool according to the ETL picker. After Topicus .Finance looked into the top results, they also decided Prefect would fit their needs. Interestingly, Pentaho Community Edition also appeared in the results, but at the bottom. This is a good sign as it shows this tool is a viable choice for their use case, but far more suitable options are available.

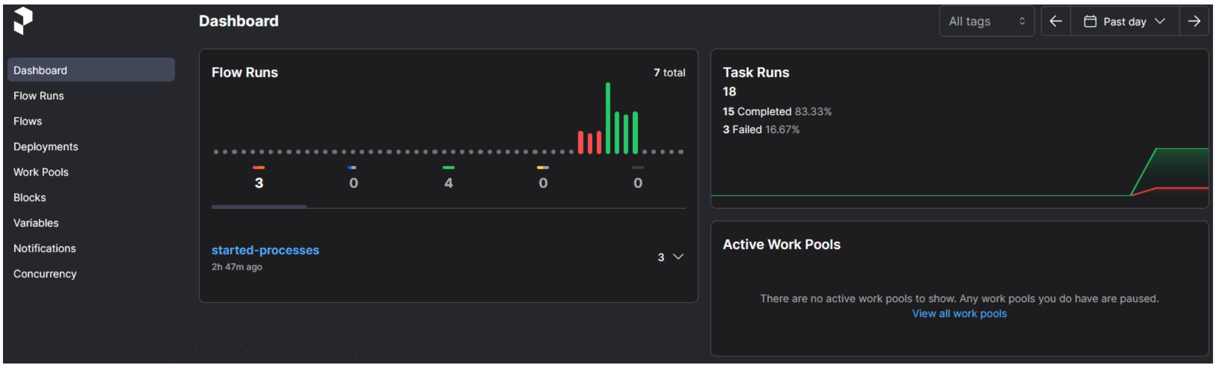


Figure 5.1: Preview of Prefect dashboard

Topicus .Finance was also interested in DBT but decided first to try out Prefect without DBT and add it in the future if necessary.

Tool	Score
Prefect	1.0
Dagster	0.9984
Mage	0.9507
Apache Hop	0.8436
Apache Camel	0.4851
DBT	0.0119
Pentaho Community Edition	0.0

Table 5.3: ETL picker results for Topicus .Finance

Prefect is a powerful orchestration tool that uses annotated Python scripts to run workflows [36]. Prefect offers different scheduling options; an extensive dashboard of flow runs and scheduled runs; thorough error logging which the user can extend; options for notification settings for a plethora of situations including when a run fails; with the available training resources Prefect is a straightforward solution that can still handle complex workflows. Furthermore, Prefect can run through Docker which Topicus also preferred.

The choice was made to create a simple flow as a test to see how Prefect works. The goal was to send a message on a specific Slack channel [1] that displays a table that summarizes the number of business lending processes that have started in the past seven days and the accumulated amount these processes are worth across all the users of Topicus .Finance's software platform. The flow consists of two parts, the first part fetches the data from the database with a SQL query. The second part posts a message on the Slack channel such that the management team can see it. With Pentaho, this was a rather complex flow to set up as Pentaho is not specifically designed as an orchestrator.

The implementation with Prefect on the other hand was much easier. First of all, Topicus .Finance preferred the implementation method of using pure Python over Pentaho's low-code building blocks. Second of all, the provided monitoring and out-of-the-box logging functionalities were praised as they were clear and straightforward and offered useful drill-down features as well as the ability to alter schedules and other settings. Furthermore, they specifically praised the ease of setting up notifications for failed runs, which can be sent to Slack, email, or practically anything else. A small preview of the dashboard is shown in figure 5.1, which shows the successful and non-successful runs of the active flows. Third of all, Topicus .Finance appreciates the ease of running everything as Docker containers.

[1] Slack is a team communication platform used by Topicus

Chapter 6

Discussion

The following sections go deeper into the results. We first discuss the implications of the results presented in chapters 4 and 5. We cover the meaning of the results and why they are useful, including several improvements for the ETL picker derived from the evaluations.

6.1 Key aspects

As mentioned the results for the business understanding were obtained in two ways, a literature study done beforehand and interviews conducted with developers. The results from the literature, found in appendix A, were more focused on key aspects of what makes a tool future-proof. Despite that, the results from the interviews with developers also overlapped. ETL design methodologies that developed as trends in the literature such as data type-based ETL processes and ensuring data quality within the ETL were also topics that the developers highlighted. However, more aspects were found in the interviews as the developers could give more insight into the important aspects when choosing a tool.

The interviews also indicated that many of the trends found in the literature are not yet as relevant in the business world as they are in the research world. Making changes in the business world is only done when the costs that have to be made to achieve these changes are worth it. So far, the biggest impact the trends from the literature have is the concept of data lakes which are starting to make their way into the corporate world. Often, a DW is preferred as it is known and often already in place. Therefore, making changes to an existing DW is much easier than creating an entire data lake on which all reports and dashboards must be rebuilt even if it will save time.

Finally, the interviews presented a more pragmatic perspective on data warehousing and ETL design compared to the theoretical approach found in the literature. Besides the literature showing novel concepts, developers care more about what they can use right now. Specifically, the information on the current version was important to developers, as they were hesitant to commit to a tool that is still very new as it has not yet proven to be a worthy contender. Therefore, the list of aspects presented in table 4.1 is more focused on results highlighted in the interviews rather than trends found in the literature.

Overall, even though the developers that were interviewed each had a different use case, the key aspects were all roughly the same. The differences lie in how a tool handles these aspects. For example, a developer who already has many of their other processes running in the cloud probably wants their ETL tool and the corresponding workflows also to be hosted in the cloud. On the other hand, if currently everything runs in docker, a new application should also be hosted in docker. In both cases hosting was important, but how the aspect was handled was far more important.

6.2 Aspect matrix & scorecard

The developed aspect matrix and accompanying scorecard show how the incorporated tools each handle the different key aspects that were previously identified. During the development of the aspect matrix, it became clear that some tools were capable of doing almost everything quite well, whereas other tools were designed for very specific tasks. This meant that certain tools would be filtered out quickly in most cases as they simply do not comply with a diverse set of use cases.

A good example of this is Airbyte, which is one of the leading tools in the field for exchanging data between two sources. However, Airbyte does not allow transformation to be made during synchronization tasks. This means this tool is only a suitable option if the users are looking into ELT and perform the transformations on request afterward. Since many use cases do require some sort of transformation during the synchronization, this tool is often not considered a viable option anymore. Conversely, Prefect and Mage are both quite capable of almost all aspects which means they are not filtered out very often. This also meant that the tools that can capably handle many key aspects are often rated higher even if more tools are still considered.

Furthermore, after analyzing the created scorecard, it became clear that two aspects are more significant in the filtering process than any other. The first is the hosting options, the second is the implementation methods the user would like. Many of the tools can be hosted using Docker, but only a few are hosted as a stand-alone application or a programming library. If a user chooses one of these latter two, the list of available tools immediately becomes limited. similarly, many tools use code or low-code implementation with scripting options as the main way to implement ETL workflows. Again, limited options are available when a user would like to implement their ETL workflows using configuration files. Combining these two strict aspects with the other filtering options can result in an empty suggestion list.

6.3 Implications of the ETL picker

Based on the quantitative results of the evaluation survey presented in table 5.2, we can see that the understandability was rated at an average of 8 with a standard deviation below 1, meaning most participants agreed that the ETL picker was understandable. Similarly, the usability was also given a high rating with an average of 7.167 and also a standard deviation below 1, which means the ETL picker by itself was at least a usable tool.

The questions were clear to the participants as they rated this with an average of 7.833, however, the standard deviation is a little higher at 1.329. This means there were some parts unclear. This was also reflected in the comments that were left, which led to some of the improvements discussed in section 6.4.

The results were rated least high with only a 5.667 on average. The standard deviation was the largest of all with 2.733, as mentioned the lowest score was a 2 and the highest a 10. Based on the comments given with these scores, we believe the lower ratings are mainly due to a misunderstanding of what the results mean. The comments left by participants included that the scoring was unclear and the results required more information about why the suggestions were suitable. The scoring mechanism is explained to the user, however, this message is not clear enough for everybody. Furthermore, the comments raised the idea not all participants are aware the results are still mere suggestions and not a final answer for them to immediately use. This is also communicated to the users by explicitly mentioning they should research the suggestions before deciding as this is still largely based on preference. The first choice might theoretically be the best but the user can still prefer any of the other suggestions for any reason.

One participant mentioned they did not get any results for one of the use cases they filled in and were surprised by this result as they would imagine that is the whole reason someone would use the ETL picker to begin with. This comment also mentioned the tool should then suggest how to simplify the use case and even suggest tools that might be suitable. This simplification is suggested by the ETL picker in the sense that the message displayed to the user gives suggestions on which questionnaire questions have the biggest impact on the result and that changing these answers would most likely lead to actual suggestions.

We deliberately chose to display a message no suitable tool was found instead of suggesting tools that might be suitable. The reason there are no results is that all tools were filtered out during the first stage of the logical model. It would require more information to determine which tools still might be suitable since this requires extra knowledge of the requirements which is not available. This would have to be added to the questionnaire as well just in case no tools are left. This would drastically increase the complexity of the ETL picker, which other participants already commented on for being too elaborate. Instead, by suggesting the user alter their answer for one of two aspects of the ETL picker that have the biggest impact on the results, the user can still receive valuable suggestions without further complicating the ETL picker.

One interesting thing to note about the general comments is that we believe most of the participants are developers with extensive domain knowledge. To clarify, the evaluation survey was anonymous and sent to multiple organizations. Therefore, it is unknown who filled in the survey exactly. However, the comments that suggested a definition list were all similar as all mentioned a definition list would clarify what the meaning of each term is in the context of the ETL picker, which might influence the way they answer the questions. This indicates that the participants are most likely developers of ETL workflows with knowledge of the domain, but this is not ideal as the decision to use a new tool most likely does not depend solely on developers. As one comment also mentions, multiple company roles are involved in setting up the requirements of a new tool. This also means perhaps the evaluation should have checked the diversity of the participants in terms of their roles.

There were also several positive comments made, one particularly interesting positive comment was made on how the ETL picker can make the user think about certain aspects of their ETL process they may not have thought of. This comment started by saying that the participants knew how to answer certain questions because they already had a solution in place. However, if someone starts from scratch and begins by filling in the ETL picker to find a tool, it forces the user to think about aspects they might not have thought of yet which will lead to better suggestions. Another positive comment mentioned that they recently also switched ETL tools and that filling in their use case yielded their final choice as a high-ranked suggestion. Overall, the ETL picker was given an average score of 6.833 with a standard deviation of 1.722. This is very promising for the first setup of such a suggestion-based tool, especially considering the difficulties the ETL picker tries to overcome.

Furthermore, the results of the case study show, to some extent, that the suggestions themselves are also not useless either. Topicus .Finance was happy with the results of the implementation with Prefect and expressed serious interest in Prefect as a complete replacement for their current tool. While this is just one example, it shows that the ETL picker can at least help certain people find a new tool, which is the core purpose of the ETL picker.

Moreover, while this study was performed in the context of ETL tools, the problem of choosing the right tool for a specific use case or the right approach to tackle a problem is broader than this. Therefore, the results obtained in this study should only be considered within the context of ETL processes and data warehousing and can not be generalized to any choice process yet. The ETL picker itself is limited in its scope because it only allows for ETL tools to be considered.

However, we do strongly believe that the results obtained show the approach taken in this study was suitable. The adaptation of the CRISP-DM methodology helps to understand the problem and helps to develop a solution. The compliance with the seven guidelines by Hevner et al. [27] created a good opportunity for the researchers to critically self-evaluate the development process to see if the ETL picker met their expectations. The survey gave insight into the usefulness and usability of the ETL picker and the case study provided an example of the ETL picker's logical model suggesting a proper tool that is indeed suitable.

6.4 Improvements of the ETL picker

Based on the results of the evaluations, several improvements can already be incorporated to enhance the ETL picker's quality. First, the entire ETL picker can be put into one app where the questionnaire is no longer separate from the results. This would allow users to alter their response and immediately see the effects. Furthermore, this allows to make the questionnaire more pleasing to look at and add extra information for certain questions behind a question mark icon to clarify certain parts. Moreover, it will overcome the limitations of a Google Form that users experienced during their evaluation. This improvement is possible in Streamlit as it does have options to store results and as already shown can work with user input and will make the entire process more streamlined. However, as is discussed in more detail in section 7.2, Streamlit does have its limitations, meaning the deployment might have to be reconsidered.

Second, the survey participants commented they would like to see more detailed explanations for the received results. Indicating the results should tell the user which of their requirements are met with an aspect comparison of the suggested tools. Although the user is notified that they should still research the suggested tools to make their final decision, giving this kind of overview would further help them make a well-thought-out decision. Furthermore, the addition of the links to the websites of each tool was requested to make this investigation step easier.

Third, the survey participants indicated they would like to see an overview of all the tools the ETL picker takes into consideration. With this, users can check if a tool they are already considering themselves is also part of the ETl picker, and thus by filling in the questionnaire they can see if this tool is suitable for their use-case.

Lastly, the survey participants indicated that several terms in the questionnaire could be misinterpreted. Therefore, it was suggested to add a definition list to the ETL picker so that users understand the meaning of each term in the context of the ETL picker. This way the users can fill in the questionnaire in a way that represents their use case.

Chapter 7

Conclusion & future work

This chapter answers the research questions and concludes this study. The research questions as defined in section 1.2 were as follows.

- **Main RQ: How can an adapted CRISP-DM methodology be used to develop a recommendation system for open-source ETL tools?**
- **Sub-RQ1:** What are the key aspects of an ETL tool for a specific use case?
- **Sub-RQ2:** How do different open-source ETL tools handle these key aspects?
- **Sub-RQ3:** How can recommendations be determined based on requirements?
- **Sub-RQ4:** How useful do users find the recommendations?
- **Sub-RQ5:** Does the adapted CRISP-DM approach result in a working recommendation system?

7.1 Answering research questions

Sub-RQ 1

The first sub-question asks about the key aspects of an ETL tool for a specific use case. To answer this question we have performed several interviews with development teams. The result of these interviews combined with one of the literature studies was a list of important aspects. The complete list is presented in section 4.1. Although all use cases and requirements were different for each team, the key aspects were almost identical. According to the teams, how an ETL tool handles these key aspects was more important.

Sub-RQ 2

The second sub-question is answered by using the key aspects found in sub-question 1 to see how different ETL tools handle each of the aspects. To answer this question, an aspect matrix for each of the tools was created by analyzing each tool and describing how each aspect was handled by each tool. Based on this aspect matrix a scorecard was created which converted these descriptions into ratings. Tools that handled aspects similarly received a similar rating. The scorecard depicts a clear representation of the strengths and weaknesses of each tool. Both the aspect matrix and scorecard are discussed in section 4.2

Sub-RQ 3

The third sub-question asks about how the recommendations can be determined. By using a questionnaire the requirements for the key aspects can be gathered from users. Using the scorecard created for sub-question 2, the ETL tools can be filtered and ranked based on the requirements specified by the users. The filtering process ensures that the tool meets all of the user's required aspects, while the ranking prioritizes recommending the most suitable tool based on its ability to perform the most critical aspects effectively. Details of the filtering and ranking process are outlined in section 4.3.2.

Sub-RQ 4

The recommendation system as a whole scored an average of 6.833 with a standard deviation of 1.722. Indicating users did find the recommendations useful but, as seen in other scores and the comments left in the survey, there is room for improvement. As discussed in sections 6.3 and 6.4, the way the recommendations are presented leaves most to be desired. The recommendation received high scores for understandability, usability, and question clarity. The survey results indicate that the recommendations are generally helpful, but they primarily lack sufficient explanation and justification.

Furthermore, the case study received highly positive feedback. The chosen ETL tool was a substantial upgrade compared to the existing ETL tool and Topicus believed the tool to be an excellent replacement as Prefect could do everything they require while being easier to use.

Sub-RQ 5

In short, the answer to sub-question 5 is yes, the adapted CRISP-DM approach resulted in a working recommendation system. This is shown by the results of the different evaluations that were conducted. The compliance with the seven guidelines by Hevner et al. [27], which was used as a self-reflection on the development process, shows that the design and implementation of the recommendation system were done satisfactorily. Furthermore, the survey and case study gave insight into the strengths and weaknesses of the recommendation system and gave insight into improvements that could be made.

Main RQ

Based on the answers to the sub-questions, we can conclude that by adapting the CRISP-DM methodology a working recommendation system can be developed for open-source ETL tools. Adapting each phase of the CRISP-DM cycle to reflect the steps of developing a recommendation system rather than a data mining model the outcome of this study was successful. The adaptations made to the original CRISP-DM allowed the researchers to gain the necessary information to develop a working recommendation system which users perceived useful. The iterative approach of CRISP-DM that was preserved allowed for improvements to be incorporated into the recommendation system in the next cycle.

Furthermore, we believe the adapted CRISP-DM methodology can be applied in many other contexts. Similar problems where different options are available as choices allow to employ the methodology used in this study. The steps to define the key aspects of the choice and how the choices handle these key aspects allow for a straightforward comparison of the options and can therefore be used for a range of other applications.

7.2 Limitations

One limitation is found in the deployment using Streamlit. Although Streamlit is an accessible platform for developing and deploying data-driven applications, participants noted inconsistencies in the app's availability during the evaluation. Since Streamlit is an open-source free-to-use tool for deploying apps, the app shuts down after a period of inactivity, resulting in users having to wait for the app to restart to find their results. While this does not impact the results of the ETL picker, it is frustrating for participants and should be addressed in future development by deploying the app differently.

Furthermore, the app showed inconsistencies in the results. It seemed as if the app used caching to store the previously displayed results even though no caching was enabled resulting in new results not being displayed. Several attempts were made to overcome this, however in the end it was not successfully fixed. The impact on the user should however be limited as the app has to restart after a period of inactivity after which the results behave as expected again.

7.3 Threats to validity

Incorporated tools

The tools incorporated in the ETL picker were found in a separate study performed before this current research. It is however possible that tools were missed during the selection process. We do not believe there to be a threat to validity for three reasons. First, the process of finding the tools is well documented and repeatable [25] and can therefore be checked by anybody. Second, the users of the ETL picker will be informed of the tools that are incorporated, as was already mentioned as one of the improvements to be made to the ETL picker. This means that if a user is missing a certain tool they might be considering they can still compare that tool to the suggestions made by the ETL picker and make a decision based on that comparison. Third and last, the ETL picker can be adapted to incorporate new tools. If the ETL picker is developed further, new ETL tools could be added to the scorecard and be taken into consideration.

Interviews

A single researcher conducted all interviews. This could lead to potential information being missed. To mitigate this the interview questions were carefully constructed with relevant topics identified beforehand both in the literature and with the help of more experienced researchers. This ensured each interview gained information on the same set of topics. To ensure all information could be extracted from the interview each interview was recorded (with permission of the attendees) such that the researcher could listen back to the answers.

Most interviews were conducted in Dutch to ensure people had no issue expressing themselves in a language they were less proficient with. One interview was conducted in English since a non-Dutch-speaking attendee was present. During this interview, the researcher noticed the Dutch attendees sometimes had trouble translating, in which case the interviewer asked them to answer in Dutch and would help translate for the non-Dutch-speaking attendees.

Furthermore, the interviews were conducted with the help of developers employed at Topicus. Therefore, it is important to see if the key aspects and evaluation results obtained in this study remain consistent when studied on a broader scale beyond Topicus. We believe this to be the case as the developers at Topicus were all working on different ETL processes that all had different requirements. Each team of developers had different concerns and found different aspects important. Therefore, we believe this is not a threat to the validity of the obtained results. One could argue with the variety of the divisions within Topicus each division can be seen as a separate company only sharing the Topicus brand.

Scorecard biases

The scorecard was created by rating each tool for each aspect based on the aspect matrix. As a single researcher conducted this process, there is a potential risk of bias, as subjective opinions could have been inadvertently incorporated into the ratings. Therefore, not all aspects were rated on the same scale. Instead, the tools were categorized into as many categories as necessary for each aspect. Tools that performed similarly would receive an equal score and tools that were categorized as better performing would all receive an equally higher score than tools that were categorized as lower performing. The difference in the number of categories was mitigated by normalizing the results every time after applying a new aspect to the score calculation.

Survey population

Unfortunately, the population of the survey evaluation was smaller than expected. This was mostly due to timing as many people were on holiday while the survey was conducted. The period in which responses were accepted was made as large as possible however the population is still small. Therefore, the quantitative results obtained from the survey should be viewed in the right context, they are promising but not yet deterministic. However, we believe the threat to validity is minimal as the survey was not entirely quantitative. The comments left by the participants gave insight into the qualities and inferiorities of the ETL picker.

7.4 Future work

The obtained results also leave room for several directions of future studies that can be performed.

7.4.1 Connectedness between tools

One comment given during the evaluation was that one tool is often not the holy grail and a complex ETL process might require multiple tools to perform all necessary tasks. The ETL picker does try to give recommendations on this with for example DBT being mentioned as well-suited if a user would like more options when transforming data and might recommend the user to look for an orchestrator to schedule their workflows if a tool does not have integrated scheduling. However, with the data presented in this study, it is not possible to deduce which tools would work well together, in what context they would work well together, and why they would work well together in said context. This will require further research into the tools and more importantly on how to define connectedness between tools in different use cases such that it allows for recommendations to be made.

7.4.2 Validate key aspects & improvements

A second aspect that should be studied in further detail is the key aspects found in the literature and the interviews. The literature study results showed various trends that emerged over the years that could become important aspects when looking at the future-proofness of a tool. Future research is needed to assess the validity and generalizability of the trends identified and utilized in this study. It is necessary to evaluate whether these trends remain as significant, have already been integrated into routine business practices, or have diminished in relevance. Additionally, future studies should examine whether the key aspects identified in the interviews continue to be decisive factors in the selection of ETL tools.

7.4.3 Inclusion of proprietary software

A third aspect of this study that grants the opportunity for further research is the incorporation of only open-source software. While this was a deliberate choice, it might be true that proprietary software is a better fit for certain use cases. At the start, we argued that open-source software is currently just as powerful as these proprietary options and this is most certainly still the case, however, proprietary software has its benefits that should be studied. Furthermore, many of these proprietary software applications are part of an iPaaS that might offer more than open-source alternatives. A future study could dive deeper into how proprietary software compares to open-source software and what aspects might make them more suitable for a certain use case as opposed to an open-source alternative.

7.4.4 Data mesh

In one of the literature studies the emergence of the data mesh was found. For ETL tools to remain future-proof, it is necessary to see if each tool is ready to be used in a data mesh architecture. In this study, the data lake and data lakehouse were incorporated, but the data mesh was omitted as it was not identified in the interviews as a key aspect. Future research should be conducted to see if the incorporated tools can be used as part of a data mesh and the recommendation system should be updated accordingly to allow the users to specify the use of a data mesh in their requirements.

7.4.5 Method validation on a broader scale

As mentioned, we believe the adaptations made to the CRISP-DM methodology for this study will also suit similar research problems. The results obtained in this study show promise for creating a recommendation system for open-source ETL tools. This raises the question if this methodology can also be applied in other contexts. This could be tested within other software domains such as a recommendation system for data storage platforms, Customer Management Systems, or any other software application. Furthermore, this methodology can be tested outside the domain recommendation systems for software, for example, in the context of recommending business or sports strategies, or materials to use for a construction project. Evaluating the applicability of the adapted CRISP-DM methodology across various contexts will demonstrate its versatility and further validate the findings presented in this study.

Bibliography

[1] Amazon. *Amazon Web Services*. 2024. URL: https://aws.amazon.com/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all (visited on 10/22/2024).

[2] J. Awiti, A. Vaisman, and E. Zimányi. "From Conceptual to Logical ETL Design Using BPMN and Relational Algebra". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11708 LNCS (2019), pp. 299–309. DOI: 10.1007/978-3-030-27520-4_21. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072985913&doi=10.1007%2f978-3-030-27520-4_21&partnerID=40&md5=5b81896b66835a6e87d0d191269ecc3e.

[3] J. Awiti and E. Zimányi. "An XML Interchange Format for ETL Models". In: *Communications in Computer and Information Science* 1064 (2019), pp. 427–439. DOI: 10.1007/978-3-030-30278-8_42. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072957643&doi=10.1007%2f978-3-030-30278-8_42&partnerID=40&md5=2ad5c4fea013823c478fead36bdf3a87.

[4] A. Behm et al. "Photon: A Fast Query Engine for Lakehouse Systems". In: 2022, pp. 2326–2339. DOI: 10.1145/3514221.3526054.

[5] Neepa Biswas, Anamitra Sarkar, and Kartick Chandra Mondal. "Efficient incremental loading in ETL processing for real-time data integration". In: *Innovations in Systems and Software Engineering* 16.1 (2020), pp. 53–61. DOI: 10.1007/s11334-019-00344-4.

[6] Neepa Biswas, Anamitra Sarkar, and Kartick Chandra Mondal. "Empirical Analysis of Programmable ETL Tools". In: *Communications in Computer and Information Science* 1031 (2019). Ed. by Mandal J.K. et al., pp. 267–277. DOI: 10.1007/978-981-13-8581-0_22.

[7] Jesús Camacho-Rodríguez et al. "Apache hive: From mapreduce to enterprise-grade big data warehousing". In: Association for Computing Machinery, 2019, pp. 1773–1786. DOI: 10.1145/3299869.3314045.

[8] Giorgio Camozzi, Felix Härer, and Hans-Georg Fill. "Multidimensional Analysis of Blockchain Data Using an ETL-based Approach". In: Association for Information Systems, 2022.

[9] Pravin Chandra and Manoj K Gupta. "Comprehensive survey on data warehousing research". In: *International Journal of Information Technology* 10 (2018), pp. 217–224.

[10] Surajit Chaudhuri and Umeshwar Dayal. "An overview of data warehousing and OLAP technology". In: *SIGMOD Rec.* 26 (Mar. 1997), pp. 65–74. ISSN: 0163-5808. DOI: 10.1145/248603.248616. URL: https://doi.org/10.1145/248603.248616.

[11] Carlos Costa and Maribel Yasmina Santos. "Evaluating several design patterns and trends in big data warehousing systems". In: *Advanced Information Systems Engineering: 30th International*

Conference, CAiSE 2018, Tallinn, Estonia, June 11-15, 2018, Proceedings 30. Springer. 2018, pp. 459–473.

[12] Gregary Dean et al. "Performance Optimization of the Open XDMoD Datawarehouse". In: Association for Computing Machinery, Inc, 2022. DOI: 10.1145/3491418.3530290.

[13] Debezium. *Debezium*. 2024. URL: https://debezium.io/ (visited on 08/16/2024).

[14] Asma Dhaouadi et al. "Data warehousing process modeling from classical approaches to new trends: Main features and comparisons". In: *Data* 7.8 (2022), p. 113.

[15] Paweł Dymora, Gabriel Lichacz, and Mirosław Mazurek. "Performance Analysis of a Real-Time Data Warehouse System Implementation Based on Open-Source Technologies". In: *Lecture Notes in Networks and Systems* 737 LNNS (2023). Ed. by Zamojski W. et al., pp. 63–73. DOI: 10.1007/978-3-031-37720-4_6.

[16] Sean Eom. "DSS, BI, and data analytics research: current state and emerging trends (2015–2019)". In: *Decision Support Systems X: Cognitive Decision Support Systems and Technologies: 6th International Conference on Decision Support System Technology, ICDSST 2020, Zaragoza, Spain, May 27–29, 2020, Proceedings 6*. Springer. 2020, pp. 167–179.

[17] Juan Espinoza et al. "Development of an OpenMRS-OMOP ETL tool to support informatics research and collaboration in LMICs". In: *Computer Methods and Programs in Biomedicine Update* 4 (2023). DOI: 10.1016/j.cmpbup.2023.100119.

[18] Yong-Liang Fang and Rong-Hua Ye. "Research and Implementation of ETL Algorithm Based on Kettle Cluster". In: ed. by Pei Z. Vol. 12331. SPIE, 2022. DOI: 10.1117/12.2652244.

[19] F. Fissore and F. Pirotti. "Migration of digital cartography to CityGML; a web-based tool for supporting simple etl procedures". In: ed. by Zlatanova S., Sithole G., and Dragicevic S. Vol. 42. 4. International Society for Photogrammetry and Remote Sensing, 2018, pp. 267–274. DOI: 10.5194/isprs-archives-XLII-4-193-2018.

[20] C. Giebler et al. "Leveraging the Data Lake: Current State and Challenges". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11708 LNCS (2019), pp. 179–188. DOI: 10.1007/978-3-030-27520-4_13.

[21] Jarrett Goldfedder. "Choosing an ETL Tool". In: *Building a Data Integration Team: Skills, Requirements, and Solutions for Designing Integrations*. Apress, 2020, pp. 75–101. DOI: 10.1007/978-1-4842-5653-4_5. URL: https://doi.org/10.1007/978-1-4842-5653-4_5.

[22] Matteo Golfarelli and Stefano Rizzi. "From Star Schemas to Big Data: 20 Years of Data Warehouse Research". In: *A comprehensive guide through the Italian database research over the last 25 years* (2017), pp. 93–107.

[23] Google. *Google Cloud*. 2024. URL: https://cloud.google.com/pricing/ (visited on 10/22/2024).

[24] Google. *Google Forms*. 2024. URL: https://www.google.com/forms/about/ (visited on 08/13/2024).

[25] J. Grotentraast. "Systematic literature review on open-source data warehouse tools and design trends". Only available through GitHub. 2024. URL: https://github.com/JGrotentraast/Research-Topics-Jurgen-Grotentraast (visited on 08/08/2024).

[26] Himanshu Gupta. "Selection of views to materialize in a data warehouse". In: *Database Theory—ICDT'97: 6th International Conference Delphi, Greece,*

January 8–10, 1997 Proceedings 6. Springer. 1997, pp. 98–112.

[27] Alan Hevner et al. "Design Science in Information Systems Research". In: *Management Information Systems Quarterly* 28 (Mar. 2004), pp. 75–.

[28] Nick Hotz. *What is CRISP-DM?* 2024. URL: https://www.datascience-pm.com/crisp-dm-2/ (visited on 08/08/2024).

[29] Voon Hou Su, Sourav Sen Gupta, and Arijit Khan. "Automating ETL and mining of ethereum blockchain network". In: Association for Computing Machinery, Inc, 2022, pp. 1581–1584. DOI: 10.1145/3488560.3502187.

[30] Søren Kejser Jensen et al. "pygrametl: A Powerful Programming Framework for Easy Creation and Testing of ETL Flows". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12670 LNCS (2021), pp. 45–84. DOI: 10.1007/978-3-662-63519-3_3.

[31] J. Kachaoui and A. Belangour. "Challenges and benefits of deploying big data storage solution". In: 2019. DOI: 10.1145/3314074.3314097.

[32] Natalija Kozmina, Laila Niedrite, and Janis Zemnickis. "Information requirements for big data projects: A review of state-of-the-art approaches". In: *Databases and Information Systems: 13th International Baltic Conference, DB&IS 2018, Trakai, Lithuania, July 1-4, 2018, Proceedings 13*. Springer. 2018, pp. 73–89.

[33] Earl Von F Lapura et al. "Development of a University Financial Data Warehouse and its Visualization Tool". In: *Procedia Computer Science* 135 (2018), pp. 587–595.

[34] Microsoft. *Azure*. 2024. URL: https://azure.microsoft.com/en-us/pricing/purchase-options/azure-account (visited on 10/22/2024).

[35] Salwa Mohammed Nejres. "Analysis of data warehousing and data mining in education domain". In: *International Journal of Advances in Computer Science and Technology* 4.04 (2015).

[36] Prefect. *Prefect*. 2024. URL: https://prefect.io/ (visited on 08/28/2024).

[37] Asma Qaiser et al. "Comparative Analysis of ETL Tools in Big Data Analytics". In: *Pakistan Journal of Engineering and Technology* 6.1 (Jan. 2023), pp. 7–12. DOI: 10.51846/vol6iss1pp7-12. URL: https://journals.uol.edu.pk/pakjet/article/view/2266.

[38] F. Ravat and Y. Zhao. "Data Lakes: Trends and Perspectives". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11706 LNCS (2019), pp. 304–313. DOI: 10.1007/978-3-030-27615-7_23.

[39] Khurram Shahzad and Jelena Zdravkovic. "A goal–oriented approach for business process improvement using process warehouse data". In: *The Practice of Enterprise Modeling: Second IFIP WG 8.1 Working Conference, PoEM 2009, Stockholm, Sweden, November 18-19, 2009. Proceedings 2*. Springer. 2009, pp. 84–98.

[40] Y. Song et al. "Design and construction of the data warehouse based on hadoop ecosystem at HLS-II". In: Joint Accelerator Conferences Website (JACoW), 2018, pp. 233–235. DOI: 10.18429/JACoW-PCaPAC2018-FRCB2.

[41] J. Sreemathy et al. "Overview of ETL Tools and Talend-Data Integration". In: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1650–1654. DOI: 10.1109/ICACCS51430.2021.9441984.

[42] Streamlit. *Streamlit*. 2024. URL: https://streamlit.io/ (visited on 08/13/2024).

[43] Madhusudhan Reddy Sureddy and Prathyusha Yallamula. "Approach to

help choose right data warehousing tool for an enterprise". In: *International Journal of Advance Research, Ideas and Innovations in Technology* 6.4 (2020).

[44] Christian Thomsen et al. "Programmatic ETL". In: *Lecture Notes in Business Information Processing* 324 (2018). Ed. by Zimanyi E., pp. 21–50. DOI: 10.1007/978-3-319-96655-7_2.

[45] John Venable. "Design Science Research Post Hevner et al.: Criteria, Standards, Guidelines, and Expectations". In: June 2010, pp. 109–123. ISBN: 978-3-642-13334-3. DOI: 10.1007/978-3-642-13335-0_8.

[46] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a standard process model for data mining". In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Vol. 1. Manchester. 2000, pp. 29–39.

[47] Yeisol Yoo and Jin Soung Yoo. "RFID data warehousing and OLAP with hive". In: Institute of Electrical and Electronics Engineers Inc., 2019, pp. 476–483. DOI: 10.1109/IUCC/DSCI/SmartCNS.2019.00105.

[48] Xiong Zhang and Wei T Yue. "Integration of on-premises and cloud-based software: the product bundling perspective". In: *Journal of the Association for Information Systems* 21.6 (2020), p. 6.

[49] Qianqian Zheng. "ETL Based Data Integration Scheduling". In: ed. by Subramanian K. Vol. 12509. SPIE, 2023. DOI: 10.1117/12.2655919.

Appendix A

Previous results

A.1 Open-source ETL tools

Tools found on web	Tools found in literature
Airbyte	Apache Druid [15]
Apache Airflow	Apache Hadoop [15, 40, 47]
Apache Beam	Apache Hive [7, 15, 29, 47]
Apache Camel	Apache Kafka [15]
Apache Hop	Apache Spark [40]
Apache NiFi	Hevo Data [41]
Apache SeaTunnel	OpenXDMoD [12]
CloudQuery	Pentaho Community Edition [18, 41, 49]
Dagster	Python libraries* [19]
DBT	R_etl [5, 6]
Keboola	Scriptella [5, 6]
Kestra	StreamSets [41]
Knime Analytics Platform	Talend [17, 41]
Mage	
Meltano	
Prefect	
PipelineWise	
Singer	

Table A.1: The complete list of tools that were found before applying the criteria. The tools that were excluded after applying the criteria are marked in red. Tools on the right were found in literature, and tools on the left were found through an accommodating web search.
*The Python libraries include: Ethereum-etl [8], Luigi, Petl [5, 6], and Pygrametl[5, 6, 30, 44]

A.2 Trends found in literature

The figures below show the categorization of trends of six main categories that were found in the systematic literature study conducted on trends [25]. For each category, the topics found in each year are put down in a table. The colors indicate which topics belong to the same trend.

DW architecture					
2018	2019	2020	2021	2022	2023
	DL: What, how, challenges, benefits, DW limitations	DW limitations	Lakehouse emergence	Lakehouse continuation	DL so far
	data quality and lifecylce	data lake platforms	Data mesh emergence	Data mesh continuation	Data mesh continuation
	metadata	Implementation	Design approaches DL	DL observations & expectations	
	textual data			DW and DL benefits and weaknesses	
Legend					
	Data Lake				
	Lakehouse				
	Data Mesh				

Figure A.1: Categorizations of trends for DW architecture

DW design					
2018	2019	2020	2021	2022	2023
Kimball	Kimball	Kimball	Kimball/hybrid		Green DW
HEFESTO					Brewer's rule
Legend					
	Kimball				
White	No trend				

Figure A.2: Categorizations of trends for DW design

Data types					
2018	2019	2020	2021	2022	2023
Integration of trajectory data	Document-oriented database/NOSQL	Semantic trajectory *	NoSQL	Graph-oriented NoSQL	Trajectory DW
Trajectory ETL with graph	Geospatial in document-oriented db	LOD			Hybrid NoSQL
LOD/semantic data	LOD	IOT			NoSQL
	IOT				IOT
Legend					
	Trajectory data				
	LOD & semantic				
	NoSQL				
	IoT data				
	Semantic trajectory				

Figure A.3: Categorizations of trends for Data types

ETL					
2018	**2019**	**2020**	**2021**	**2022**	**2023**
Big data ETL	BPMN for ETL	Near real time ETL for big data	Metadata ETL	Dynamic ETL	
NoSQL	Quality assurance/data validation	Quality metrics of ETL	User-generated content ETL	Data cleaning	
Variety of data	Data mining	Specific ETL tool metadata based	Near real time ETL		
Ontology based ETL		Data quality	Semantic ETL		
Cleaning			Virtual DW		
Near real time ETL					
Legend					
	Data type-based ETL				
	Data quality				
	Near real time ETL				
	Metadata-based ETL				
White	No trend				

Figure A.4: Categorizations of trends for ETL

Performance					
2018	**2019**	**2020**	**2021**	**2022**	**2023**
Data model for predictable execution time	Dynamic data placement	Decoupling data management and computation	Data access/joining algorithm	Physical design through data mining	Cost-based optimizer
	Physical design/partitioning	Cost-based optimizer	Materialized views	Graph-oriented framework	Decentralized cluster
	GPU-based BJISP		Big data integration	Physical design general	Divided ETL
			PatchIndex		
Legend					
	Data placement & partitioning				
	Table join optimization				
	Query-plan optimizer				
White	No trend				

Figure A.5: Categorizations of trends for performance

Schema design					
2018	**2019**	**2020**	**2021**	**2022**	**2023**
Temporal DW	Data vault	Data vault	Data vault	Schema generation from natural language	Data cube (hyper lattice)
Automatic schema evolution	Temporal DW	Schema evolution from queries	Semi-automatic schema design	ML-based measure detection	Temporal graph cube **
Volunteer design	Ontology-based design	Schema design for big data	Ontology-based schema generation	Temporal DW	
Data cubes	Schema from document-oriented DB	Ontology-based schema generation			
Security design in the cloud	post-mining generalized association rules	Schema comparisons			
	Data cubes	Hybrid design methodology			
	Dynamic structure				
Legend					
	Temporal DW				
	Schema detection, generation & evolution				
	Data Cube model				
	Data Vault model				
	Ontolog-based design				
	Combines ontology-based with schema generation				
	Combines Data Cube with temporal data				
White	No trend				

Figure A.6: Categorizations of trends for schema design

Appendix B

Interview questions developers

1. **General**
   (a) Who are you? What is your background? What does your team do?
   (b) How is your current ETL tool being used?
      i. Is it used for internal use or as part of an external service for clients?
   (c) What are the shortcomings of this current ETL tool?
      i. Are things missing?
      ii. Is the functionality not useful/not fitting for your use case?
2. **ETL specific**
   (a) What do your ETL pipelines look like?
   (b) How are these designed?
   (c) How do you guarantee data quality in your pipelines?
   (d) How are the pipelines started?
      i. Is there a scheduler?
      ii. Are jobs being run in parallel?
         A. How does that work?
   (e) How do you ensure security in your pipelines?
      i. Are you working with a lot of sensitive data?
      ii. How secure is your hosting?
   (f) Why is the current ETL tool no longer suitable for your needs?
   (g) What would an ideal situation of design, scheduling/triggering of pipelines, parallelism, and security look like with a new ETL tool?
3. **Version control**
   (a) How important is version control for your team?
      i. Do you work with different versions of your ETL pipelines for different clients?
      ii. In what cases do old versions need to be restored?
   (b) How are changes to pipelines reviewed?
   (c) What are the problems in the current way of version control?
   (d) How would version control and change reviews ideally be done?
4. **Quality checks**
   (a) How are pipelines tested?

(b) What are the problems with the current way of testing pipelines?
(c) How would this be done ideally?

5. **Closing**

(a) Some tools are novel and have not “matured” fully yet, what is your view on these upcoming tools? Would you consider using them?
(b) Are there any other topics or points of interest we have not discussed yet?

Appendix C

Survey questions

Introductory text:
This evaluation survey is designed to evaluate the ETL picker, a framework designed to help choose a new open-source ETL tool. This evaluation is part of a graduation assignment at the University of Twente. The answers to this evaluation are completely anonymous and are only used to improve the working of the ETL picker.

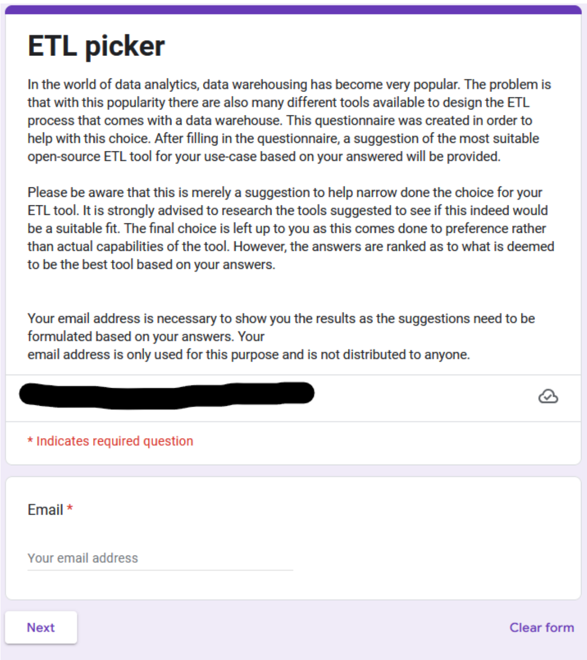
Please take a look at the ETL picker and answer the questions below afterward. You can fill in the ETL picker as many times as you like to answer the questions. Please fill it in multiple times with different scenarios in mind to get a grasp of how different scenarios result in different suggestions.

Questions and type of answer:

1. On a scale of 1-10, how easy is the ETL picker to understand? (rating 1-10)
2. What makes it easy/difficult to understand? (open question)
3. How do you rate the usability? Think about the way the ETL picker is presented to you and how it works (rating 1-10)
4. What could be better in terms of usability? (open question)
5. How clear are the questions that were asked? (rating 1-10)
6. If anything, what was unclear about them? (open question)
7. Were there questions or answers missing? (open question)
8. How clear were the results? (rating 1-10)
9. How can the results be improved? (open question)
10. Was there anything else missing? Or could anything else be improved? (open question)
11. What is the overall score you would give the ETL picker? (rating 1-10)

Appendix D

Questionnaire



ETL picker

In the world of data analytics, data warehousing has become very popular. The problem is that with this popularity there are also many different tools available to design the ETL process that comes with a data warehouse. This questionnaire was created in order to help with this choice. After filling in the questionnaire, a suggestion of the most suitable open-source ETL tool for your use-case based on your answered will be provided.

Please be aware that this is merely a suggestion to help narrow done the choice for your ETL tool. It is strongly advised to research the tools suggested to see if this indeed would be a suitable fit. The final choice is left up to you as this comes done to preference rather than actual capabilities of the tool. However, the answers are ranked as to what is deemed to be the best tool based on your answers.

Your email address is necessary to show you the results as the suggestions need to be formulated based on your answers. Your
email address is only used for this purpose and is not distributed to anyone.

[redacted]

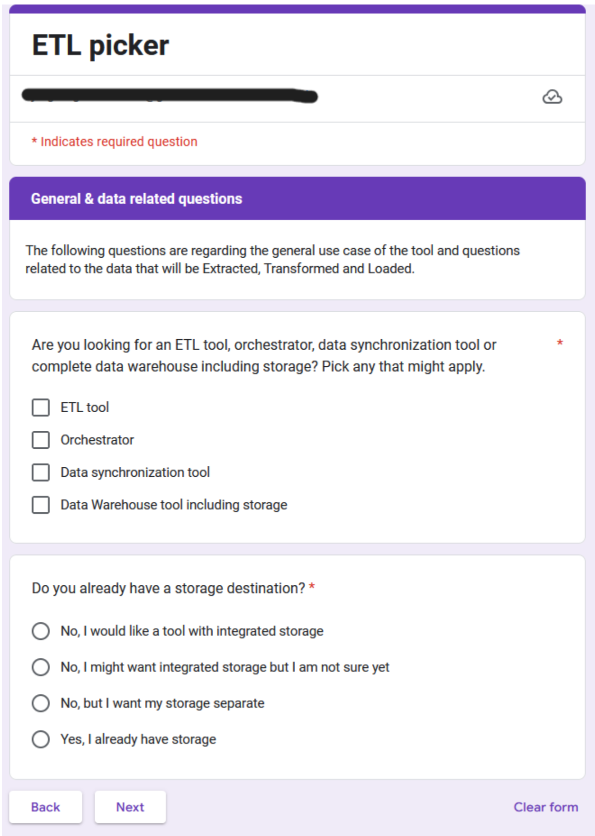
* Indicates required question

Email *

Your email address

Next

Clear form



ETL picker

[redacted]

* Indicates required question

General & data related questions

The following questions are regarding the general use case of the tool and questions related to the data that will be Extracted, Transformed and Loaded.

Are you looking for an ETL tool, orchestrator, data synchronization tool or complete data warehouse including storage? Pick any that might apply. *

- ☐ ETL tool
- ☐ Orchestrator
- ☐ Data synchronization tool
- ☐ Data Warehouse tool including storage

Do you already have a storage destination? *

- ○ No, I would like a tool with integrated storage
- ○ No, I might want integrated storage but I am not sure yet
- ○ No, but I want my storage separate
- ○ Yes, I already have storage

Back

Next

Clear form

ETL picker

[redacted]

* Indicates required question

Data

These questions are related to the type of storage that would suit your use case. You see these questions because you did not know yet what kind of tool you are looking for or because you are looking for included storage.

Do you need to combine data from many different (types of) sources? *

- ◯ Yes, I have many different sources
- ◯ I have a few different sources
- ◯ No, I only have one or two sources

What type of sources do you have? *

- ☐ Database(s)
- ☐ File(s)
- ☐ API(s)
- ☐ Specific application(s)

How much does this data need to be transformed in order to fit your needs? *

- ◯ The data needs to undergo various and complicated transformations
- ◯ The data needs to undergo simple transformations
- ◯ The data is stored raw as is

How often does the source or destination schema change? *

- ◯ (Very) often
- ◯ Sometimes
- ◯ Rarely/not at all

How will your data be stored? *

- ◯ Structured
- ◯ Unstructured
- ◯ Both
- ◯ In another application (e.g. CRM or SCM systems)
- ◯ Don't know yet

How often does new data need to be loaded in? *

- ◯ Near real time
- ◯ Every hour
- ◯ Every half day
- ◯ Every day
- ◯ Less than once a day

Is the data size too large to drop and refill the entire table every time? *

- ◯ Yes, data is too large so only updates and newly inserted data should be captured
- ◯ No, refilling the entire database at once everytime is okay

Back Next Clear form

ETL picker

[redacted]

* Indicates required question

Technical architecture & security

The following questions are regarding the technical architecture the new tool will be hosted in as well as any resource configurations you might want to do and security.

How would you like to host the application? *

- ☐ A docker container
- ☐ Stand alone application
- ☐ Programming library
- ☐ Cloud hosting

If you are considering cloud hosting, what kind of cloud provider would you like to use for running your ETL processes? Please leave blank if you are not considering cloud hosting

- ◯ I already have separate cloud provider/I want a separate cloud provider
- ◯ I would like the application to offer a (payed) cloud version

What minimum resource configuration requirements do you have? *

- ◯ Yes, I need full control of how much resources individual parts can use
- ◯ I need to know how much resources my pipelines use and atleast set a maximum
- ◯ I only need to know how much resources were used afterwards
- ◯ I don't have any requirements

If resource configuration is done through config files, what type of configuration files would you like to use? Leave empty if not applicable

- ☐ HOCON
- ☐ JSON
- ☐ XML
- ☐ YAML

Do you already have security in place for hosting and running your ETL securely or do you want a tool to help you with that? *

- ◯ I already have security in place
- ◯ I have some security but would like the tool to have options for securing my pipelines
- ◯ I don't have security yet and want security options in the new tool
- ◯ I don't have security yet but will implement this without the new tool

Are you working with a lot of sensitive data which needs to be masked or encrypted? *

- ◯ Yes
- ◯ No

Back Next Clear form

ETL picker

Log in bij Google om je voortgang op te slaan. Meer informatie

* Verplichte vraag

Implementation

The following questions are regarding if you want to implement your ETL pipelines using programming languages or not.

How do you prefer to implement your ETL pipelines *

- [ ] Only code
- [ ] No code blocks with scripting possibilities
- [ ] Configuration files
- [ ] Pure no code blocks

If you want to use programming or scripting, what programming language(s) do you want to code in? Leave empty if not applicable

- [ ] C#
- [ ] C++
- [ ] Groovy
- [ ] Java
- [ ] Javascript
- [ ] Python
- [ ] R
- [ ] Ruby
- [ ] Scala
- [ ] Shell
- [ ] SQL
- [ ] Anders:

Vorige | Volgende | Formulier wissen

ETL picker

Log in bij Google om je voortgang op te slaan. Meer informatie

* Verplichte vraag

Monitoring & Scheduling

The following questions are regarding monitoring and scheduling needs.

How important is monitoring for your use-case? *

	1	2	3	4	5	
Not important	○	○	○	○	○	Very important

How extensive monitoring is required? *

	1	2	3	4	5	
Basic error logging	○	○	○	○	○	Full dashboard with drill down capabilities

What type of scheduling do you want? *

- [ ] CRON/time based schedule
- [ ] Event triggers
- [ ] Trigger other workflows from within a workflow

Can scheduling be done with another tool? *

- ○ Yes
- ○ No

Vorige | Volgende | Formulier wissen

ETL picker

[redacted]

* Indicates required question

Version control, community & learning

These last questions are regarding version control, the community that uses the tool and the amount of learning resources needed

How important is version control? *

	1	2	3	4	5	
Not important at all	○	○	○	○	○	Very important

How important is a strong community? *

	1	2	3	4	5	
Not important	○	○	○	○	○	Very important

How important is training and onboarding of the new tool? This includes documentation, (video) tutorials and other guidelines. *

	1	2	3	4	5	
Not important	○	○	○	○	○	Very important

Send me a copy of my responses.

Back | Submit | Clear form

Appendix E

Streamlit code

```
streamlit.title("ETL Picker")
streamlit.write("Thanks for using the ETL picker!")
url = "https://forms.gle/xzdqHWCDZSXCN9YG6"
streamlit.write("If you have not done so please first fill in the questionnaire on which this tool depends through this [link](\%s)" \% url)
email = streamlit.text_input("Please fill in your email address to see your results")
if streamlit.button("See results"):
```

Figure E.1: Streamlit code for creating the first page of the ETL picker

The code above shows how to add a button and several pieces of text to a Streamlit app. If a method should be called at the push of a button, all that is needed is to write *"if streamlit.button("text"):"* and within the if statement the method that should be called. When running the app, a button with the text will be displayed. Text input can be added to an app by using *"streamlit.text_input()"*. Any text can be written to the app using *"streamlit.write("text")"*. More methods are available for creating styling elements like a title or subtitle and there are specific methods for writing certain data types like dataframes to ensure these are properly displayed. With only the six lines of code shown in E.1, the first page of the ETL picker front end as shown in figure 4.1a is created and the input can be used as it is immediately assigned to a variable.