In: Gröger, C., Schwarz, H. und Härder, T. (Hrsg.) Datenbank-Spektrum, Band 23, Heft 2. Springer Nature Switzerland AG, pp. 123–132, 2023. This is the author's version of the work. It is posted at https://opencms.uni-stuttgart.de/fak5/ipvs/departments/as/publications/stachch/dasp_23_barents.pdf for your personal use. The final publication is available at Springer via 10.1007/s13222-023-00443-4.

A Recommender Approach to Enable Effective and Efficient Self-Service Analytics in Data Lakes

Christoph Stach[1*], Rebecca Eichler[1] and Simone Schmidt[1]

[1]IPVS, Universität Stuttgart, Universitätsstraße 38, Stuttgart, 70569, Deutschland.

*Corresponding author(s). E-mail(s): Christoph.Stach@ipvs.uni-stuttgart.de;
Contributing authors: Rebecca.Eichler@ipvs.uni-stuttgart.de;

Abstract

As a result of the paradigm shift away from rather rigid data warehouses to general-purpose data lakes, fully flexible self-service analytics is made possible. However, this also increases the complexity for domain experts who perform these analyses, since comprehensive data preparation tasks have to be implemented for each data access. For this reason, we developed BARENTS, a toolset that enables domain experts to specify data preparation tasks as ontology rules, which are then applied to the data involved. Although our evaluation of BARENTS showed that it is a valuable contribution to self-service analytics, a major drawback is that domain experts do not receive any semantic support when specifying the rules. In this paper, we therefore address how a recommender approach can provide additional support to domain experts by identifying supplementary datasets that might be relevant for their analyses or additional data processing steps to improve data refinement. This recommender operates on the set of data preparation rules specified in BARENTS—i.e., the accumulated knowledge of all domain experts is factored into the data preparation for each new analysis. Evaluation results indicate that such a recommender approach further contributes to the practicality of BARENTS and thus represents a step towards effective and efficient self-service analytics in data lakes.

Keywords: Data Lake, Data Preparation, Data Pre-Processing, Data Refinement, Recommender, Self-Service Analytics

1 Introduction

For years, data warehouses were considered the primary data infrastructure when it comes to data analytics. For well-defined analysis purposes (e.g., reports), data are collected from heterogeneous sources, pre-processed, and stored in a uniform format in the data warehouse. The ETL process which handles these tasks is generally implemented by a data engineer since this requires not only data knowledge but also a great deal of IT expertise. As all types of analyses that have to be supported are known in advance, it is possible to specify once, how the data have to be prepared. Once the data are prepared and stored in the data warehouse, domain experts can use online analytical processing to autonomously define queries that extract the knowledge they need [12].

Yet, these straightforward self-service analytics come at the price of severely limited flexibility. The available raw data are tailored to the pre-defined use cases before they are made available in the data warehouse, which means that information content is lost and thus precludes other use cases a priori. Moreover, the analysis tools are usually tailored to specific purposes, such as the creation of reports, and can only be parameterized to a limited extent. To enable more dynamic data analytics, data lakes were introduced. In contrast to the data warehouse, the ETL process is modified in such a way that data are extracted from the sources and loaded into the data lake as raw data—the transform step only

takes place when the data are accessed (also known as ELT). This requires analysts, however, to implement data preparation themselves each time the data are used. That includes use-case independent tasks, such as data cleansing, as well as use-case dependent tasks, such as format conversion and schema transformation [11]. To reduce the redundant implementation of data preparation tasks, data lakes provide not only raw data but also variants of these data in different pre-processing stages. In a zone architecture, e.g., zones dedicated to specific use cases can be created, in which data are prepared for precisely this use case [8]. Yet, effective self-service still requires an IT expert as domain experts generally do not have the necessary IT skills to implement the required data preparation tasks [14].

Therefore, we introduced a data preparation toolset for data lakes called BARENTS[1] [17]. It consists of two parts, an ontology-based model that allows domain experts to specify data preparation tasks as transformation rules, and a processing engine that applies such rules to the targeted data. Due to its ontology-based approach, it is possible to implement a plugin for existing graphical ontology editors (e.g., similar to CoModIDE[2]) to provide an intuitive user interface when working with BARENTS. Yet, domain experts face two problems in this process. First, they reinvent the wheel over and over again as the data preparation tasks they require may have already been specified for another use case. Second, they are not made aware of all the available data that could enrich their analyses.

In this paper, we deal with these problems by introducing a recommender approach for BARENTS. The foundation for this is the set of rules for data preparation specified in the BARENTS ontology—i.e., the collected knowledge of all domain experts. We discuss different types of recommendations that can support domain experts in their work and implement the most promising combination of these recommendation types to improve the practicality of BARENTS.

The remainder of the paper is organized as follows: In Section 2, we outline BARENTS and its ontology, insofar as it is required for this work. Then, in Section 3, we discuss research approaches to support domain experts in self-service analytics. We introduce our recommender approach for BARENTS in Section 4. Subsequently, we assess whether this approach makes self-service analytics more effective and efficient in Section 5. Section 6 concludes this work.

[1]BARENTS stands for tailora**b**le d**a**ta p**re**paration zo**n**e for da**t**a lake**s**.

[2]see https://comodide.com/ (accessed on May 10, 2023)

2 BARENTS

Unlike data warehouses in which rigid data schemas ensure a well-defined internal structure, data lakes require some sort of structural organization and curation of the data. There are many different approaches to this, with zone architectures prevailing in literature [8]. Zone architectures have in common that data extracted from external sources and loaded into the data lake sequentially transit a series of zones. Each of these zones represents a processing stage of the data. The zones can be divided into two groups: some zones are relevant for all users of the data lake (e.g., a zone in which all raw data are stored), while others are intended for a specific type of use (e.g., a refined zone in which the data are prepared according to a strict schema similar to the data warehouse) [15]. These two types of zones thus divide a data lake into two distinct areas: a use-case-independent area, consisting of a fixed set of pre-defined general-purpose zones, and a use-case-dependent area, in which the data are tailored to the intended purpose [8]. Since a data lake is supposed to support any kind of data analysis, the latter area has to be highly dynamic and extendable by further zones dedicated to new use cases if needed. The transition between these two zones is therefore a significant challenge for self-service analytics since domain experts initially have to transform the generically prepared data from, e.g., the raw data zone to meet their requirements.

It is generally presumed that a data scientist performs this task. An idealistic assumption is that s/he has domain knowledge with insights into what has to be analyzed, data knowledge with an understanding of how the data have to be prepared for this purpose, and IT knowledge to implement all of this. Yet such a versatile skill set is hard to find. While domain experts have an understanding of the domain and the data, they lack IT knowledge. For IT experts, it is the other way around. To solve this dilemma, domain experts must be empowered to prepare the data for their use cases as autonomously as possible, thus ultimately enabling self-service analytics for data lakes.

BARENTS provides a solution to reduce the IT requirements necessary to specify data preparation tasks. For this purpose, BARENTS introduces an ontology that can be used to define data processing rules. Such a rule consists of three parts: a source from which the data originate, a processing operator that is to be applied to the data, and a sink in which the result is to be stored. Figure 1 illustrates a generic processing rule.

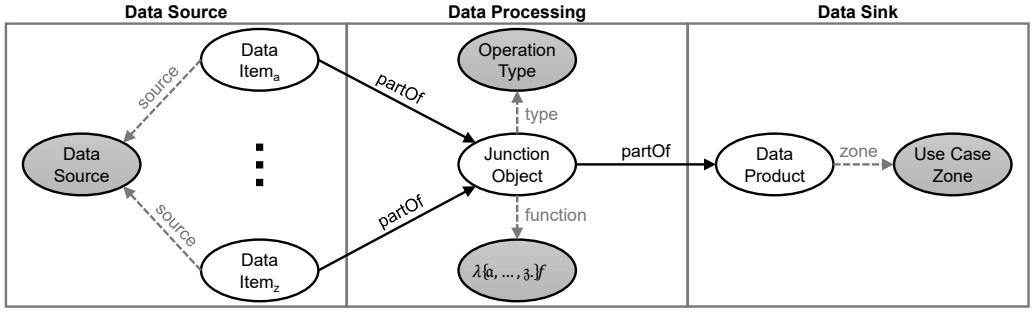

Figure 1 An Exemplary Instance of a Data Processing Rule Specified in the BARENTS Ontology [16].

When selecting a data source, it is also necessary to specify which data items are affected by the processing rule. This is mandatory as heterogeneous data are stored in a common zone. An internal structuring, similar to tables in a database, cannot be presumed.

The selected data items represent the input for a processing operator. These operators are adopted from the functional programming paradigm, namely the map operator (i.e., an operator that applies a unary function to all data items), the filter operator (i.e., an operator that evaluates a unary predicate logical expression for each data item and drops all items for which it evaluates to false), and the reduce operator (i.e., an operator that aggregates all data items to a single result value using a binary function). Using these operators from functional programming has two advantages: First, they can be composed at runtime and applied to a stream of data items easily. Second, their program logic can be defined using plain lambda expressions. Implementing such expressions can be handled by domain experts since their syntax is quite simple and resembles mathematical formulas. In addition to these three operation types, arbitrary user-defined functions can also be specified, although this requires advanced programming skills.

The output of these operators is stored in a new data lake zone. However, since this output may be needed only temporarily, BARENTS introduces the concept of virtual use case zones. Virtual use case zones hold data only temporarily, similar to a data buffer. Such zones are particularly needed when processing rules are required that consist of several processing operators. For this purpose, a data sink can serve as the data source of another processing rule. In this way, arbitrarily complex processing rules can be composed, which consist of many individual source-processing-sink segments.

BARENTS uses a Python-based processing engine that parses this ontology, i.e., the sum of all defined processing rules, and applies them to the corresponding data. The access to data sources and data sinks is handled by adapters. In BARENTS, we use the readers and writers of pandas[3], which provide access to different data infrastructures. The use of pandas has another advantage, as the data processing can easily be applied to the pandas DataFrame that contains the data. This processing engine conceptually constitutes a new data lake zone, which realizes the data transit from the use-case-independent area into the use-case-dependent area [16]. In the context of this work, however, it is only relevant that such an ontology exists, which consists of many processing rules that have been defined for a data lake.

Its strong focus on data preparation in data lakes distinguishes BARENTS from other data science platforms such as KNIME[4]. Here two assumptions can be made, namely that the zones of a data lake are known in advance and that the actual data analysis takes place outside the data lake in domain-specific dedicated tools. As a result, the operators in BARENTS are lightweight (e.g., data are accessed via predefined adapters, so only the appropriate zone needs to be specified, without the need to provide technical parameters for access) and reduced to the essentials of data preparation. So, domain experts are not overwhelmed by an overload of information and options and can concentrate on their actual tasks. For more information about BARENTS, please refer to our previous work, e.g., Stach et al [17].

Although BARENTS enables domain experts to prepare their data autonomously and thus facilitates self-service analytics in data lakes, it is also important that the specification of this ontology can be done effectively and efficiently. To ensure that the specification can be done **effectively**, a domain expert must be informed of both available data and data refinement steps that may be relevant to his/her use case. To ensure that the specification can be done **efficiently**, it is crucial that

[3]see https://pandas.pydata.org/ (accessed on May 10, 2023)
[4]see https://www.knime.com/ (accessed on May 10, 2023)

domain experts do not always have to reinvent the wheel and can make use of existing processing rules.

In the following section, we discuss research approaches that aim to provide this kind of support.

3 Related Work

A review of the scientific literature reveals four main research directions that deal with the support of domain experts in the preparation of data in data lakes. As a data lake contains data on a wide variety of areas and at different processing stages, the first research direction is dealing with facilitating the retrieval of relevant data. A second research direction aims at identifying data available in the data lake and related to data already selected by a domain expert. A third research direction tries to recommend data cleansing and data transformation steps appropriate to the data. Finally, there is also a fourth research direction that seeks AI-based fully automated data preparation. We discuss these four research directions in more detail in the following. Due to the large number of research papers in these areas, we only discuss representative examples for research direction.

Retrieval of Relevant Data. The initial step of any analytics task is the selection of appropriate base data. To support domain experts operating on data lakes, one approach to this problem is to bring some sort of order to the plethora of available data. For this purpose, there are different data partitioning models that group the data into different partitions, e.g., based on structural or semantic similarities [10]. This enables domain experts to specifically access a partition that is suitable for their purposes. Metadata help to find relevant data or partitions. In addition to the data origin, these metadata can also describe inherent data features such as quality or completeness. Furthermore, approaches for automatic keyword extraction can be used to describe entire datasets based on a few keywords. By integrating all these metadata into a unifying metadata model, a data catalog can be created, which domain experts can query to find data of interest for their use cases [5].

Identification of Related Data. Once such a structure is established for a data lake, even more comprehensive support can be provided for finding data for an analysis task. By enriching the metadata with, e.g., information regarding the similarity between datasets, data engineers can analyze relationships among the datasets. In this way, domain experts can also be referred to available data with similar content or supplementary data to their selected base data [9]. Other approaches aim at gathering extensive metadata concerning the data schema. Especially with respect to data preparation, similar datasets based on schema-matching metrics can be identified for the domain expert. All of these datasets can then be pre-processed in a similar manner [2]. By using data mining on the combined metadata, domain experts can therefore be comprehensively supported not only in the retrieval of relevant data but also in the identification of data that are related to their use cases.

Recommendation of Processing Steps. After having gathered all data required for an analytics task, the domain expert has to decide how to prepare them appropriately. There are a few approaches that support the selection of the appropriate processing steps as well. Similar to the metadata about the data, some metadata about their pre-processing is also included in a data lake. This serves primarily the purpose of data provenance. For instance, metamodels can be used to document which basic operators, such as formatting, calculation, and join, were applied to which data. However, for base data with comparable characteristics, these metadata can also be used to identify applicable processing operators based on their previous usage [13]. However, there is much less research work on this topic than on finding suitable base data. This is remarkable since process mining provides a comprehensive toolset for analyzing and comparing business processes [1]. So, data preparation processes could be mined in a similar way if described in a machine-processable format.

Fully Automated Data Preparation. The most advanced research direction aims to fully automate data analysis. This applies in particular to the data selection and data preparation part. By analyzing historical data regarding data preparation processes, machine learning models can be trained and then applied to new tasks. These models thereby replace the domain expert entirely [4]. While this sounds promising as domain knowledge can simply be imported via the models, studies show that the involvement of domain experts leads to much better results. Therefore, instead of excluding human involvement, humans should become an integral part of the analytics loop. That is, the main research focus should be on providing them with comprehensive tool support [3].

Synopsis. As fully automated data preparation is not an option, domain experts need tool support to handle self-service analytics in data lakes. While there are several approaches for the identification and retrieval of relevant data that facilitate these tasks, there is a lack

of comprehensive solutions, especially when it comes to the recommendation of appropriate data processing operators. This is partly due to the fact that no extensive metadata exist regarding the data preparation. Such a knowledge base, however, would be a fundamental prerequisite for a recommender. As the BARENTS ontology not only contains such information but is also machine-processable, we discuss in the following section how these facts can be leveraged to make data preparation more effective and efficient.

4 A Recommender for BARENTS

When domain experts use BARENTS to create data processing rules, they can be supported by a recommender in various ways. We have identified four concepts that can be applied to this end, which are outlined below.

Concept A. In analogy to Megdiche et al [13], heuristics can be established for which data source, what data processing operators are eligible. This can be done based on the number of selected data sources—e.g., if two sources are selected, only binary operators, such as a join, are eligible—as well as based on their characteristics—e.g., on a data source for unstructured text data, arithmetic operators are not applicable. Using such heuristics, the number of possible data processing operators can be severely limited immediately after the selection of data sources. This kind of recommender does not require any additional knowledge (besides the heuristics). This is both a curse and a blessing. The advantage is that the recommender can support domain experts right from the start since it does not need historical data for training. However, therein also lies the major crux of this approach. As the domain knowledge persisted in the BARENTS ontology is taken into account, the recommendations are only very generic. This can slightly improve the useability, but domain experts do not get any in-depth support.

Concept B. To accomplish this, data and domain knowledge must be applied. This knowledge is available in the form of the BARENTS ontology. Based on the selection of a data source, a recommender could query the ontology for rules that have already been defined for this (or a similar) source. As a result, the result set of the recommender is a subset of the recommendations of Concept A. Unlike Concept A, however, not just any applicable data processing operator is suggested, but rather those that have turned out to be useful for this data source in previous use cases. Thus, these are actual recommendations, whereas Concept A only excludes operators that are technically not feasible. Yet, a domain expert is not only informed about suitable operators but also about sinks in which the data s/he needs are already available in a refined state. As each sink can be a source for a new data processing rule, s/he can then select this sink and see further recommendations for additional data preparation steps. S/he can also select rules from the recommendations and adapt them (e.g., change the parameters of a lambda function) and thus add new rules to the ontology.

Concept C. As the BARENTS ontology represents a graph consisting of all processing rules—keep in mind that each end node of a rule (i.e., a sink) can be used as a start node for a subsequent rule (i.e., a source)—it is also possible to perform comprehensive graph queries on it. A recommender can take advantage of this fact by retrieving subgraphs that are similar to the set of rules a domain expert has defined for his/her use case so far. While in Concept B only incrementally single data preparation steps can be suggested, Concept C is therefore able to recommend entire chains of data processing operators. Furthermore, additional intermediate steps can be suggested, which the domain expert had not thought of. The corresponding rules can then be inserted in his/her subgraph. However, this concept requires that the domain expert has already defined a sufficient number of processing rules for his/her use case in order to be able to make meaningful recommendations. Thus, this type of recommender is particularly suitable for refining an existing rule base.

Concept D. Many recommender systems use collaborative filtering to further refine the recommendations and tailor them to individual users. To this end, the behavior of a user is initially analyzed. Based on this analysis, the user base is clustered, with users within a cluster being as homogeneous as possible, while users from two different clusters are as heterogeneous as possible. In the recommendations, mainly the historical data of users from their cluster are taken into account. In the context of BARENTS, however, this approach has two crucial issues. One technical problem is that BARENTS does not support linking users to their rules. While this problem would be relatively easily solved by extending the BARENTS ontology to include user information, there is a conceptual problem as well. Collaborative filtering inevitably ensures that users only get recommendations from their bubble. Thus, a domain expert would never get fresh input on how the data could be prepared in a more target-oriented way in order to achieve a better result.

Findings. In the context of BARENTS, Concept A is not very effective, since existing prior knowledge (in the form of the ontology) is completely ignored. Yet, this is essential since it contains the data and domain knowledge that is decisive for the success of data preparation. Furthermore, Concept A also only generates a superset of the recommendations of Concept B. Since the latter can produce much more tailored recommendations, we do not use the heuristic-based Concept A for our recommender. Besides Concept B, Concept C also seems promising to us since it can suggest entire chains of data processing operators as well as alternatives to existing chains. Concept B and Concept C complement each other quite well, since Concept B allows a fine-grained step-by-step rule generation, while Concept C operates with patterns consisting of many single rules and can also make recommendations on how to optimize existing rules. Concept D, however, is counterproductive as domain experts are never made aware of data preparation patterns other than those to which they are already accustomed.

While Concept B and Concept C can refer to data sources that contain the results of data processing rules (i.e., that were used as sinks), it is not possible to refer to new data that have not yet been included in any rule in the ontology. In order to refer to such data, additional metadata on the data lake itself is required. However, since metadata management is necessary regardless of BARENTS to keep a data lake operable, we do not address this issue in this paper. For more information on how to implement such metadata management, we refer to the preliminary work of Eichler et al [6].

Implementation. Thus, a combination of Concept B and Concept C is needed for BARENTS. For this purpose, it is necessary to mine the BARENTS ontology. The ontology is stored as an RDF/XML file[5]. For our recommender, we use RDFLib[6] to read and parse the file and to build a traversable RDF graph. Via a query interface, programs (e.g., an editor used by domain experts to create new data processing rules) can interact with the recommender. Depending on whether a query is received for a data source or a rule graph, it is forwarded to one of two processing modules in which Concept B and Concept C, respectively, are implemented.

The implementation of Concept B is pretty straightforward. It is only necessary to find rules that include a data source that matches the one in the query. In this context, 'matches' refers either to a perfect match or to a high degree of similarity, e.g., with respect to the features of the provided data—the more metadata available, the more fine-grained the matching can be. The rules for which there is such a match are returned in RDF/XML format as a recommendation. If the domain expert chooses one of these rules, a new query is sent to the module looking for rules that use the sink of the chosen rule as a source.

The implementation of Concept C requires more advanced graph processing capabilities. For this, we use NetworkX[7]. In order to find chains of data processing operators that are similar to the given subgraph, we convert it to a NetworkX graph. We systematically add and remove source-processing-sink triples to / from this graph and check whether the resulting graph is contained in the ontology. To this end, we compute the graph edit distance. The graph edit distance (GED) describes the similarity between two graphs:

$$GED(g_1, g_2) = \min_{(e_1,\ldots,e_k) \in P(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$

Let g_1 and g_2 be the two graphs in question. $P(g_1, g_2)$ denotes a set of k operators $e_1, \ldots, e_k$ required to transform g_1 into a graph isomorphic to g_2. Finally, $c(e_x)$ is a function that assigns to each transformation operator specific costs. In other words, the GED thus represents the minimum cost incurred in converting g_1 to g_2.

If this value is below a given threshold (i.e., if there is a sufficient similar chain of data processing operators in the ontology), the corresponding section from the ontology is included in the result set. While the entire ontology can be analyzed for relatively small subgraphs, this creates too much overhead for large subgraphs. Therefore, in these cases, the search space has to be restricted in advance using heuristics.

In the following section, we assess whether our recommender approach for BARENTS enables domain experts to perform effective and efficient self-service analysis in data lakes.

5 Discussion

To assess the practicality of our recommender in self-service analytics, and thus to evaluate how it increases the effectiveness and efficiency of working with

[5] see https://www.w3.org/TR/rdf-syntax-grammar/ (accessed on May 10, 2023)

[6] see https://rdflib.dev/ (accessed on May 10, 2023)

[7] see https://networkx.org/ (accessed on May 10, 2023)

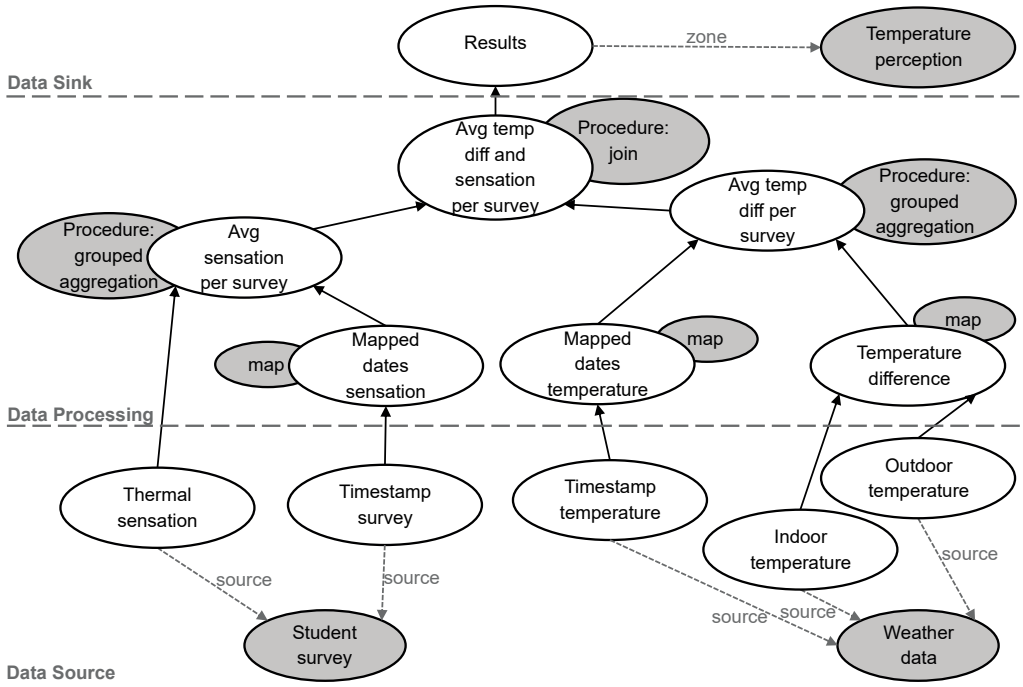

Figure 2 Pertinent Excerpt from the BARENTS Ontology Related to Leading Question 1 (Simplified and Abridged Depiction).

BARENTS, we look at a real-world use case introduced by Gao et al [7]. In this use case, a wide range of data was collected over an extended period of time using sensors and surveys in a private school environment in a suburb of Melbourne, Australia. These data include outdoor and indoor weather data (e.g., humidity and temperature), environmental factors (e.g., noise level and carbon dioxide concentration), student health data (e.g., skin temperature and heart rate), and physiological data (e.g., engagement and emotion). The purpose of this data gathering is to identify the effects of external factors on students' learning behavior and mental state in order to create an optimal learning environment.

We adopted this dataset and the authors' leading questions as the basis for our assessment. For this purpose, we have selected three different analyses, which should be processed with the help of BARENTS. As the recommender requires historical data in terms of a comprehensive ontology, we presume in our assessment that the processing of Leading Question 1 and Leading Question 2 has already been modeled by domain experts in BARENTS. The processing of Leading Question 3 shall be created from scratch. We examine how the recommender provides support in this process.

Leading Question 1: *Does a disparity between indoor and outdoor temperature affect how people perceive indoor temperature?*

To address this question, the ontology excerpt shown in Fig. 2 was created. It has to be noted that for the sake of better readability, a simplified depiction is used here and some intermediate steps are omitted.

For this question, it is necessary to combine weather data (namely indoor and outdoor temperature) with the thermal sensation derived from the survey data. To this end, the timestamps initially have to be converted to a uniform time format by means of a map operator, since the timestamps of the surveys are coarse-grained categories ('morning', 'noon', and 'afternoon'), while the temperature timestamps are given as actual points in time. The temperature difference between indoor and outdoor temperature can also be computed using a map operator. Both types of data are then aggregated on a daily basis and the mean values are joined. The results are stored in the 'temperature perception' use case zone of the data lake.

Leading Question 2: *Is there a gender difference with respect to temperature perception?*

To address this question, the ontology excerpt shown in Fig. 3 was created—again, a simplified and abridged representation is used in the figure.

In the base data used, the student information is kept separately from the survey results. Therefore, these data first have to be merged. For this purpose, a join can be implemented via the participant ID. Gender splitting

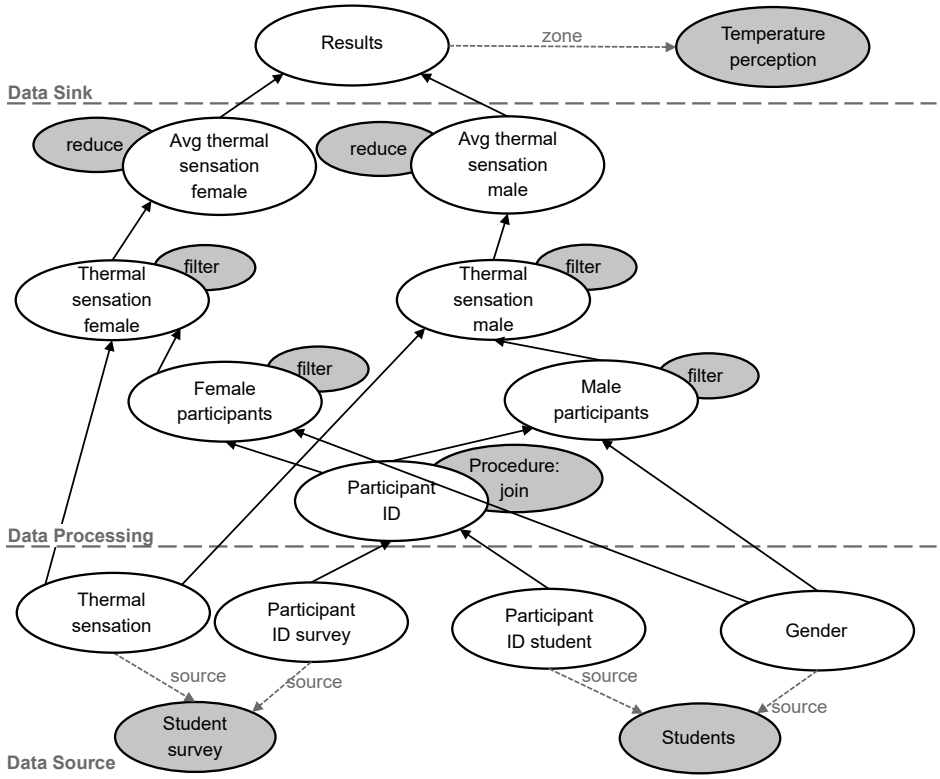

Figure 3 Pertinent Excerpt from the BARENTS Ontology Related to Leading Question 2 (Simplified and Abridged Depiction).

is realized using two filter operators, which filter out all survey results from male and all survey results from female participants, respectively[8]. Subsequently, only the thermal sensation data from the questionnaires are retrieved by means of another filter operator from these two intermediate result sets. Finally, the mean thermal sensation for both groups is computed with a reduce operator and also stored in the 'temperature perception' use case zone.

Leading Question 3: *Does the gender-specific temperature perception have any ramifications with regard to Leading Question 1?*

Let us assume that the results of the two previous analyses are available and that Leading Question 2 has revealed that gender-specific temperature perception among students can be observed. Motivated by these findings, another domain expert wants to investigate the impact on Leading Question 1. S/he starts by selecting the data source with the gender information. Our recommender can then suggest the two filter operators so that s/he does not need to implement them again. Based on the resulting subgraph, the recommender is able to retrieve the operator chain that merges these data with the survey data (see Leading Question 2). The domain expert can also reuse this chain for his/her analysis. Likewise, after selecting the weather data as a data source, the domain expert can incrementally obtain the rules for computing the temperature differences between indoor and outdoor temperatures. Yet, when merging these two branches, the domain expert has to make some adjustments. Since the survey results are now split into male and female results, two separate aggregations have to be done. The results are also stored in the 'temperature perception' use case zone.

However, the domain expert made a mistake in this analysis because s/he forgot to adjust the timestamp in the survey data—keep in mind that s/he adopted this subgraph from Leading Question 2, which did not require such adjustments. Our recommender can also assist the domain expert in fixing this issue, as it can identify the corresponding subgraph from Leading Question 1 using graph edit distance and recommend the required modifications. The resulting (simplified and abridged) ontology excerpt is shown in Fig. 4.

[8]Note that in these base data, it is assumed that there are two genders, only. This does not reflect the authors' opinion, nor is it intended to be discriminatory.

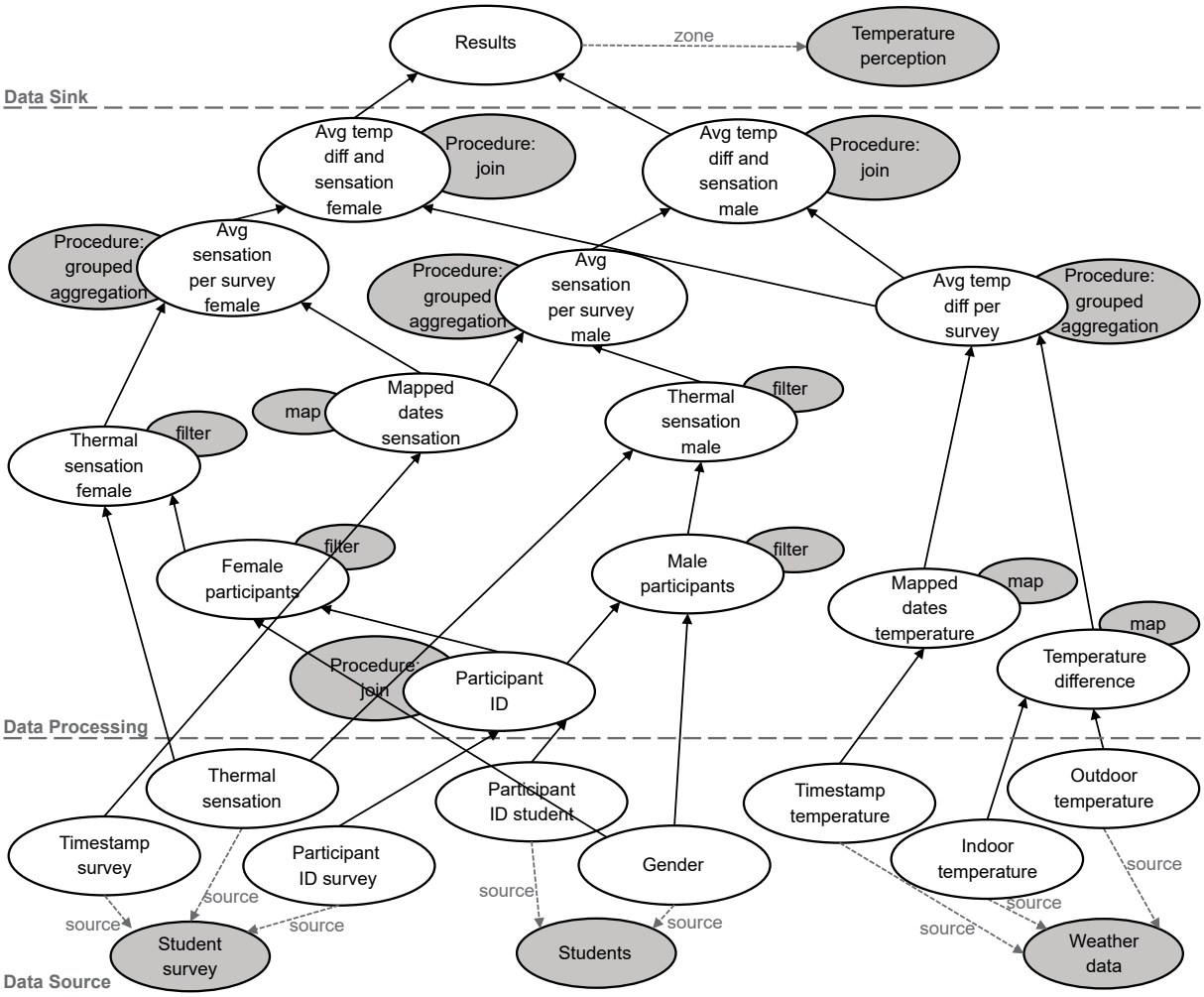

Figure 4 Pertinent Excerpt from the BARENTS Ontology Related to Leading Question 3 (Simplified and Abridged Depiction).

Yet, it has to be mentioned that the number of recommendations largely depends on the extent and scope of the ontology. With a more complex ontology, the process would not be as straightforward as in this example. It would involve several iterations with intermediate stage and query refinement steps. The general principle, however, remains the same.

Lessons Learned. As can be seen in this use case, BARENTS enables domain experts to autonomously retrieve the required data, prepare them, and transform them as needed for the intended analyses. Thus, BARENTS enables self-service analytics in data lakes. However, as Leading Question 1 and Leading Question 2 illustrate, specifying the data processing rules for these simple analyses entails a considerable amount of work. Therefore, support is needed so that data preparation can be performed effectively and efficiently.

Leading Question 3 demonstrates that our recommender can make domain experts aware of data refinement steps that further increase the data utility, e.g., in the case of the omitted timestamp adjustment. It can also refer to additional data sources if they are frequently merged with other sources used by the domain expert, e.g., in the case of joining gender data with survey data. Yet, for a more comprehensive data source recommendation, e.g., based on semantic features, additional metadata about the sources are needed, which by design are not included in BARENTS. Still, Leading Question 3 indicates that working with BARTENTS is more **effective** thanks to the recommender.

This use case also shows that, when specifying new rules, the recommender can refer to data preparation rules that are already part of the ontology, so

that they can be reused. Both single rules and even extensive subgraphs can be recommended to the domain expert. It is also possible to subsequently tailor the suggested rules to individual requirements. Therefore, working with BARTENTS is more **efficient** thanks to the recommender as well.

6 Conclusion

The foundation of any data analysis is data refinement. To enable self-service analytics, it is therefore a prerequisite that data preparation can be carried out autonomously by domain experts without extensive IT knowledge. This is particularly difficult in the case of data lakes, as data transformation must be implemented prior to data access. Using BARENTS, the required data processing can be specified as simple data preparation rules, which are then applied to the data. To support domain experts in this rule specification, we discussed different recommender concepts in this paper and implemented the most promising one for BARENTS. With this recommender, it is possible to refer to additional source data and processing rules that increase data utility. Moreover, the recommender facilitates reusing previously specified rules. A real-world use case demonstrated that the recommender makes working with BARENTS more efficient and effective.

Funding. Open Access funding enabled and organized by Projekt DEAL.

References

[1] van der Aalst W (2012) Process Mining: Overview and Opportunities. ACM Trans Manage Inf Syst 3(2):7

[2] Alserafi A, et al (2020) Keeping the Data Lake in Form: Proximity Mining for Pre-Filtering Schema Matching. ACM Trans Inf Syst 38(3):26

[3] Behringer M, et al (2020) Empowering Domain Experts to Preprocess Massive Distributed Datasets. In: BIS'20, pp 61–75

[4] Brazdil P, et al (2022) Automating Data Science. In: Metalearning: Applications to Automated Machine Learning and Data Mining. Springer, Cham, p 269–282

[5] Diamantini C, et al (2021) An Approach to Extracting Topic-guided Views from the Sources of a Data Lake. Inform Syst Front 23:243–262

[6] Eichler R, et al (2020) HANDLE - A Generic Metadata Model for Data Lakes. In: DaWaK'20, pp 73–88

[7] Gao N, et al (2022) Understanding occupants' behaviour, engagement, emotion, and comfort indoors with heterogeneous sensors and wearables. Sci Data 9:261

[8] Giebler C, et al (2020) A Zone Reference Model for Enterprise-Grade Data Lake Management. In: EDOC'20, pp 57–66

[9] Halevy A, et al (2016) Goods: Organizing Google's Datasets. In: SIGMOD'16, pp 795–806

[10] Hlupić T, et al (2022) An Overview of Current Data Lake Architecture Models. In: MIPRO'22, pp 1082–1087

[11] Inmon B (2016) Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump. Technics Publications, Basking Ridge

[12] Inmon WH (2005) Building the Data Warehouse. John Wiley & Sons, Inc., Indianapolis

[13] Megdiche I, Ravat F, Zhao Y (2021) Metadata Management on Data Processing in Data Lakes. In: SOFSEM'21, pp 553–562

[14] Michalczyk S, et al (2020) A State-of-the-Art Overview and Future Research Avenues of Self-Service Business Intelligence and Analytics. In: ECIS'20, p 46

[15] Sharma B (2018) Architecting Data Lakes. O'Reilly Media, Inc., Sebastopol

[16] Stach C (2023) Data Is the New Oil–Sort of: A View on Why This Comparison Is Misleading and Its Implications for Modern Data Administration. Future Internet 15(2):71

[17] Stach C, et al (2021) Demand-Driven Data Provisioning in Data Lakes: BARENTS — A Tailorable Data Preparation Zone. In: iiWAS'21, pp 187–198