SmartDataLake

DELIVERABLE D4.1

Interactive visual analytics model

PROJECT NUMBER: 825041
START DATE OF PROJECT: 01/01/2019
DURATION: 36 months

Horizon 2020

SmartDataLake is a Research and Innovation action funded by the Horizon 2020 Framework Programme of the European Union.

The information in this document reflects the authors' views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/her sole risk and liability.

Dissemination Level	Public
Due Date of Deliverable	Month 16 (30/04/2020)
Actual Submission Date	27/04/2020
Work Package	WP4: Scalable and Interactive Visual Analytics
Tasks	Task 4.1: Visual analytics model
Type	Report
Lead Beneficiary	University of Konstanz
Approval Status	Submitted for approval
Version	1.0
Number of Pages	33
Filename	SmartDataLake-D4.1-Interactive_visual_analytics_model.pdf

Abstract

This report presents the model for the visual analytics component, defining entities and their interactions involved in the visual data analysis task. More specifically, it indicates how the user is included in the knowledge generation and sensemaking loop, refining parameters for the automated lower-level components of SmartDataLake using interactive interfaces and visualizations.

History

Version	Date	Reason	Revised by
0.1	14/01/2020	Table of Contents	Thilo Spinner
0.2	07/04/2020	First version for internal review	Thilo Spinner
0.3	14/04/2020	Revised version	Thilo Spinner
1.0	27/04/2020	Final version for submission	Thilo Spinner

Author List

Organization	Name	Contact information
UKON	Thilo Spinner	thilo.spinner@uni-konstanz.de
UKON	Michael Blumenschein	michael.blumenschein@uni-konstanz.de
ARC	Dimitris Skoutas	dskoutas@athenarc.gr
RAW Labs	Benjamin Gaidioz	ben@raw-labs.com

Executive Summary

This deliverable describes the Visual Analytics Model, which defines the interfaces and the interplay between automated components of the SmartDataLake analysis pipeline and human sensemaking.

Starting with an introduction to data analysis (Section 1) and the general concept of Visual Analytics (Section 1.1), the significance of including the human in the loop of knowledge generation is emphasized. Core ideas of Visual Analytics as well as key terms are introduced, which build the baseline for the description of the Visual Analytics Model (Sections 1.1.1 to 1.1.4).

After this general introduction, the concept of Visual Analytics is transferred and applied to the concrete components and tasks of SmartDataLake (Section 1.2). The role of the three major tasks of WP4, namely Data Profiling (Section 1.2.1), Parameter Optimization (Section 1.2.2), and Results Visualization (Section 1.2.3) is elaborated based on the three main components of SmartDataLake, SDL-Virt, SDL-HIN, and SDL-Vis. Distinct use-cases and examples from these three components demonstrate the practical relevance of Visual Analytics for all stages in the analysis workflow of SmartDataLake. Finally, the individual components are placed into context with the project outline and the current state of SmartDataLake by connecting the Visual Analytics Model to the related Work Packages (Section 1.3).

Following the introduction, the theoretical foundation for SDL-Vis is described by the abstract Visual Analytics Model for SmartDataLake (Section 2). Starting with an embedding of the Visual Analytics Process as defined by Keim et al. [1] into SmartDataLake, each of the entities data, model, and visualization is connected to the components of SmartDataLake (Sections 2.1.1 to 2.1.3). For each of these entities, a detailed problem statement is given, illustrating the challenges of SmartDataLake and how the concepts and mechanisms of the presented Visual Analytics Model addresses them. Concluding the Visual Analytics Process in SmartDataLake, the relationships between data, model, and visualization are explained (Sections 2.1.4 and 2.1.5). This gives an overview of the connections between those entities and how human understanding and sensemaking affect all of them equally and mutually. Extending the Visual Analytics Process by Keim et al. [1] to the Knowledge Generation Model for Visual Analytics by Sacha et al. [2] (Section 2.2), gives a more detailed view on the steps of human sensemaking in Visual Analytics, building a baseline for action and reaction patterns implemented and referred to in the following sections.

After specifying the theoretical foundation of the Visual Analytics Model of SmartDataLake, the SDL-Vis architecture and the concrete implementation are explained, building the instantiation of the Visual Analytics Model. SDL-Vis is split into frontend, the Visual Explorer (Section 2.3), and backend, the Visual Analytics Engine (Section 2.4). Following a detailed discussion on the interaction patterns and design choices implemented in Visual Explorer, exemplary user interfaces, targeted towards a specific task or use-case of SmartDataLake, are elaborated in the form of Visualization and Interaction Panels (Section 2.3.2). The examples give insights in how the Visual

Analytics Model is concretely implemented, how automated components and users interact, and which parameters and results are involved in the analysis process. Finally, the Visual Analytics Engine is motivated from the needs for a computational backend, and its implementation details are discussed.

A conclusion summarizes the content of this deliverable and, in particular, the Visual Analytics Model and its role in SmartDataLake (Section 3).

Abbreviations and Acronyms

API	Application Programming Interface
BI	Business Intelligence
CSS	Cascading Style Sheets
DOM	Document Object Model
HIN	Heterogeneous Information Network
HTML	Hypertext Markup Language
IT	Information Technology
JSON	JavaScript Object Notation
KDD	Knowledge Discovery in Databases
REST	Representational State Transfer
SDL	SmartDataLake
SQL	Structured Query Language
t-SNE	T-distributed stochastic neighbor embedding
UI	User Interface
VA	Visual Analytics
WP	Work Package

Table of Contents

1. Introduction

The increasing storage and computation capacity of modern information technology systems has led to a massive amount of data being collected and stored. Virtually every action taken by individuals, automated algorithms or machines produces data, which is then stored in a variety of different formats, such as tables, text, images, and others. With modern hardware resources, the amount of data that can be collected by far outgrows the human capacity for the analysis of that data.

This information overload often leads to missed opportunities, since the knowledge hidden in the data might be relevant for personal, political or entrepreneurial decisions. For example, pilot partners of SmartDataLake have specialized on the consolidation and analysis of data on companies. A best-possible sensemaking of this data is highly relevant for the customers of those partners, since they might be the foundation for wide-ranging business decisions. If meaningful information is lost or remains unnoticed, this means competitive disadvantages for these companies. Other pilot partners focus on the analysis of financial markets. Since the benefits of an advanced analysis of those markets often is only marginally better than an average performance, the business model is dependent on an ideal exploitation of all available information. Often, the collected data is neither stored in a single location, nor in such a way that it can be combined easily. Hence, to make sense of the collected data there are two important steps, necessary for a sense making process:

First, different heterogeneous data sources must be merged and (pre-)processed in a unified fashion. This is a highly complex task, since data might be either structured, semi-structured or unstructured, contain varying attributes or be of strongly varying quality. For example, data records in different data sources often describe the same real-world entity but might have a limited set of attributes shared. Therefore, entity resolution is one relevant sub-problem in the analysis and sensemaking pipeline of SmartDataLake.

Second, the huge amount of merged and pre-processed data needs to be analyzed to extract relevant knowledge which can then be used during data exploration, for knowledge generation and within the decision-making processes. Analyzing large amounts of data is a highly complex task as scalability issues and the correct choice of mining parameters affect the useful analysis process.

A core part of these analysis tasks are automated, machine-learning-driven data processing pipelines which support analysts in merging and analyzing the underlying data. Configuring and optimizing such pipelines requires, however, a deep understanding of the underlying data, the analysis tasks, and the needs of the analysts. Given that the automated algorithms are configured appropriately with useful parameters, they can output results according to the understanding and the knowledge of their creators, but they are not able to actually make sense and create knowledge out of their own results. Configuring fully automated analysis processes and feeding in appropriate parameter settings is a challenging task as it depends on the amount of data and their characteristics. Hence, a trial and error process is often applied until useful settings have been identified. Finally, sense-making often presupposes specific domain knowledge or requires the iterative refinement of the analysis question during the analysis process in an explorative manner [1].

1.1. Visual Analytics

This is where *Visual Analytics* [2] comes into play. Instead of having the user interpret the results of an otherwise static data processing and visualization pipeline, Visual Analytics enables the user to combine data characteristics, automated data analysis using machine learning models and data mining techniques, visualizations for visual data exploration, and the knowledge generation process within a unified and interactive model which we call *interactive visual analytics model*. Figure 1 shows a simplified representation of the different components and how they related to each other. We summarize these components in the following, as they are described by Sacha et al. [3].

1.1.1. Data

Before starting the analysis, data has to be collected, selected and transformed. Data can be of structured, semi-structured, or unstructured form. The analysis goal and the available data have to feature a connection to reveal useful insights, otherwise it is unlikely to generate meaningful findings during the analysis or, in the worst case, spurious relationships might be discovered [4]. The data collection and pre-processing steps are essential for Visual Analytics and often have a significant influence on the quality of data and, consequently, the analysis results. Furthermore, it is common that data is pre-processed and augmented during the analysis, for example using clustering or classification techniques. The origin of data plays an important role for assessing the trustworthiness of analysis results. Therefore, provenance tracking [5] often is an essential requirement in large data processing and -analysis pipelines.

1.1.2. Models

Models are any form of descriptions of data. Simple descriptive statistics are an example for a model of low abstraction, while a complex data mining pipeline is an example for a model of high abstraction. Generally, the KDD process is about building models from data. These models can be used in different ways during the Visual Analytics process. Verification of an existing hypothesis might be possible with a simple model, computing a single number which either supports or rejects the hypothesis. More complex models, for example patterns extracted by data mining methods, can be used for the visual exploration of a dataset and for the creation of new hypotheses. Furthermore, automatically created models can be visualized to gain an understanding about the data abstraction process itself.

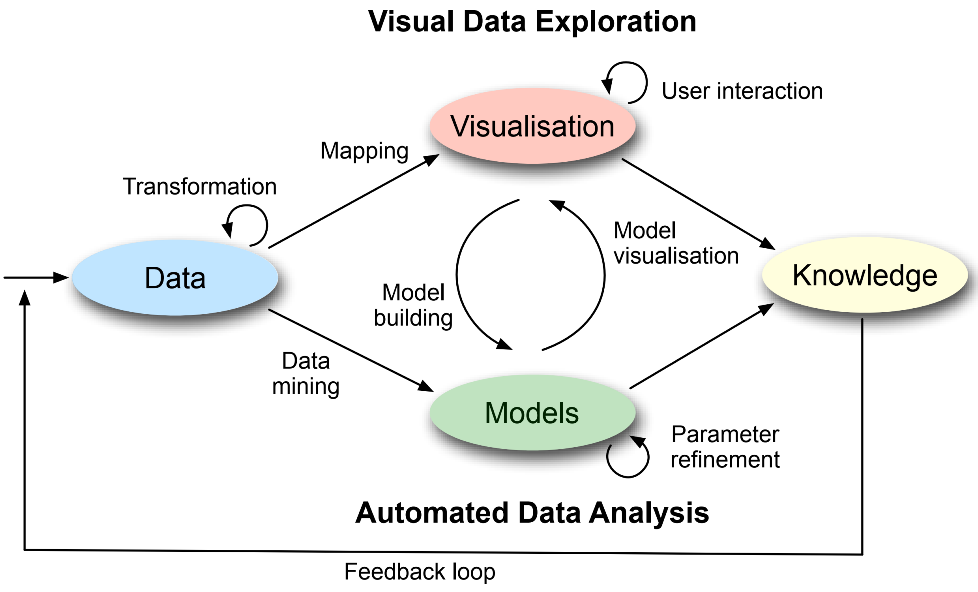


Figure 1: General Visual Analytics Model by Keim et al. [2].

1.1.3. Visualization

While knowledge might directly be generated from models, for complex tasks visualization often is essential. Both, models and data can be visualized to enable the analyst to detect meaningful relations in the data that might otherwise be overlooked. Visualizations are often used to visualize the results of automated models, for example the visualization of clusters generated by automated clustering algorithms. In Visual Analytics, the state of a visualization (e.g., viewport, filters, selections) often determines the parameters for underlying models. For example, different properties of a model might be visualized during semantic zooming. While models can be interpreted directly for knowledge generation, visualization facilitates the understanding of these models and their results. Therefore, visualizations are often used as the main interface between user and automated visual knowledge generation pipelines.

1.1.4. Knowledge

The analysis of data typically starts with one or more analysis question. Furthermore, the analysis process might be influenced by the analyst's prior knowledge about the problem domain, the data domain, or data analysis in general. The goals of an analysis process are:

1. building hypotheses about relations that are existent in the data (in the exploration loop),
2. verifying or discarding existing hypotheses (in the verification loop),
3. and, finally, generating knowledge from these hypotheses (in the knowledge generation loop).

Therefore, knowledge generation in Visual Analytics is about the verification of existing assumptions and the generation of new hypotheses in the data domain. The evidences and knowledge discovered during the analysis process enable the analyst to reason about assumptions made on the problem domain. Furthermore, discovered evidences might have different qualities, which allows the analyst to rate the trustworthiness of the generated knowledge. This not only affects evidences on the results of models, but also on the models themselves. For example, the result of a transparent statistical measure might be trusted more than the output of a complex data mining pipeline with multiple abstract parameters involved. The analyst has to find either enough evidence to trust a hypothesis and substantiate it as gained knowledge or discard the hypothesis and return to the exploration of new, undiscovered connections in the data. All steps in the Visual Analytics pipeline, from data gathering over data mining to data visualization have to be critically assessed and understood to substantiate insights and generate knowledge. In return, knowledge can be generated about all entities in the analysis pipeline: the quality of data, the connections hidden in data, the involved models, their parameters, and the Visual Analytics process itself.

The main advantage of visual analytics is that it unifies classical data processing, machine learning and visualization in a unified and tightly coupled model. Gained knowledge in the analysis process is fed back into the pipeline and can help to adjust parameter settings, pre-process and filter data, and change visual representation to highlight particular findings in more detail. This tight coupling between user and data processing is achieved by not only presenting the actual outputs of the mining processes to the user, but also information about the processes themselves. Visualizations both show results, as well as information on how they were generated. Simultaneously, real-time interaction techniques enable the user to adapt parameters of the underlying data analysis pipeline based on the insights generated by the presented information. This integration of the human in the analysis process combines the strengths of both human and machine in a single and interactive model.

The described general visual analytics model has been generalized and extended in various directions, for example by Sacha et al. focusing the knowledge generation part [3] and to understand how uncertainty of the data, the model, and the users is incorporated in the model [6].

While Visual Analytics is considered an interdisciplinary field covering many research areas, such as visualization, data mining, data management, data fusion, statistics and cognition science [2], a particular Visual Analytics Application is typically highly specialized to the domain, data collection, and task it targets. The pipeline of data loading, data mining and data visualization involves a wide variety of design and parameter choices dependent on the available data, the field of application, and the analysis goal. Therefore, those parameters, as well as the results, must be reflected in the application as understandable and accessible as possible. This, in return, affords highly specialized user interfaces, visualizations, and interaction techniques.

To cope with the complexity of both interfaces and presented data, user guidance often is an essential part of Visual Analytics applications. In the form of recommender systems, it supports the analyst in the process of data exploration and knowledge generation by "suggesting previously unseen yet potentially relevant information" [7]. Furthermore, user guidance is relevant for the understanding and refinement of abstract parameters that are to be set in the data processing pipeline. The interpretability of abstract data, therefore, may directly influence the trustworthiness and quality of the analysis results. For example, indicating the outcome of a potential parameter change prior to the change actually being applied can provide the analysist with an intuitive understanding on how the parameter influences the results while at the same time preserving real-time analysis [8].

1.2. Role of Visual Analytics in SmartDataLake

Work Package 4 (WP4), *Scalable and Interactive Visual Analytics*, strives to *involve data scientists in the data analysis and sensemaking process*, allowing the orchestration and optimization of the automated lower-level components of SmartDataLake. Enabling the interactive loop between automated and human parts of the complex data processing, data mining, and data analysis pipeline of SmartDataLake, *Visual Analytics* is the core concept of this work package. This deliverable focuses on describing the theoretical description of the Visual Analytics Model used in the SmartDataLake project, the concrete implementation of the backend and frontend, and how the different modules relate to each other (Task 4.1 to Task 4.4). Furthermore, we will report our results of the feature exploration and parameter tuning analysis tools which we already started implementing, and which are planned in the near future (Task 4.2).

The final goal of SmartDataLake is to *combine* and *analyze* large-scale multiple *heterogeneous data sources*. This complex task involves many components, ranging from data access over mining algorithms to interactive data exploration. We will implement a variety of Visual Analytics tools to support analysts in the analysis and exploration process. These tools will be one of the core features to generate knowledge out of the data lake.

In the following, we give a short overview about the role of visual analytics within core tasks addressed by the project.

1.2.1. Data Profiling

Data profiling refers to the task of automatically determining metadata about a dataset. It is a common and important activity of virtually any practitioner or researcher working with data, especially when confronted with new datasets for which little or no knowledge is available a priori. Its applications include data exploration, database management and reverse engineering, data integration, data cleaning, and data analytics, compression and governance [9].

One of the major challenges in data lakes is that data are often not accompanied by any useful metadata, thus having unknown structure, content and quality [10]. To start a successful data analysis or to combine several heterogeneous data sources together, it is essential to first obtain an overview of the underlying data. To understand the general structure of the data helps ultimately to select the most appropriate analysis tools, choose a good initial parameter set, and help to interpret the results of automated and interactive approaches.

Getting an overview of data is a multifaceted process. For example, analysts wish to understand the quality of their data. How many missing values are in the data? Are there any duplicates?

Another important process during data profiling is to analyze the distribution of data attributes in order to find out whether the data is skewed and/or follows a normal distribution. This is important to know such that correct statistical or machine learning based analysis can be performed. If the data is, for example, skewed, then pre-processing needs to be applied first.

To get a general understanding of the data, Visual Analytics can support analysts successfully. Different charts and analysis tools are necessary to get a complete picture of the data. However, choosing an appropriate technique often requires domain knowledge, depends on the data characteristics, etc. Visual Analytics can help to automatically propose the useful visualizations for a given analysis task. Furthermore, it can guide the analysis by an iterative workflow in which the human, visualizations, and automatic algorithms work together. The core focus hereby is that we do not have different small tools for different analysis perspectives, but that everything is combined in a unified framework – both from an implementation and a theoretical point of view. This helps to take the findings of one analysis step and directly inject it in the next without importing or exporting the data and parameters.

1.2.2. Parameter Optimization

The majority of components in the SmartDataLake project depend on user-defined parameters influencing the quality of the final results. It is, therefore, essential to communicate these parameters to the user to enable reliable and well-founded decision making. Furthermore, the correct choice of parameters is heavily influenced by the analysis task and might not be immediately apparent. Therefore, the user not only needs to be presented with the parameter choices but should also be able to refine parameters and explore the available parameter space and the corresponding analysis results interactively.

The problem becomes even more challenging, considering that most of these parameters are somewhat abstract. For example, entity resolution (i.e., the merging of different data entities which represent the same real-world entity) naturally involves automated decisions on whether two entities should be merged. The sensitivity of this trigger, namely the *entity resolution threshold*, has to be carefully explored. If it is chosen wrong, relationships between seemingly independent entities might be missed, or, alternatively, independent real-world entities might be merged erroneously. Having to optimize parameters without being able to examine the influence on the results makes the analysis process a time-consuming and tedious task, probably even leading to a loss of essential information hidden in the analysis pipeline.

Visual Analytics is essential to resolve these issues and enable a well-founded decision making based on the analysis results. Intermediate results can be shown to the analysts with the help of sophisticated visualizations, along with the current parameter set. Then the user can interactively modify the parameters which triggers the next analysis iteration, which is then, again, represented by the visualization. Furthermore, meta-visualizations can be used to represent the impact of specific parameters. The core advantage of Visual Analytics (compared to standard analysis processes) is hereby the tight integration between the parameters, the model to compute the analysis results, and domain knowledge of the human, which are all tightly coupled within the visual analytics model.

1.2.3. Results Visualizations

The results of a large-scale analysis in data lakes is typically very complex and difficult to understand for analysts. Often, both the input data and the analysis results are high-dimensional – meaning they contain a large number of dimensions and records. Furthermore, the data and results may have a spatial context (e.g., the geolocation of companies), may change over time (e.g., if we analyze using a sliding-window approach), or are arranged in a network structure (e.g., the heterogeneous information network that the entirety of data in SmartDataLake builds).

Interactive visualizations can be of great benefit to interpret the results and validate whether the findings are reliable and ultimately make sense. The knowledge of the user and output of the algorithms play a vital role in the general understanding process. Therefore, it makes sense to directly link the result visualizations with the input data, and the output and parameters of the analysis models. Hence, findings in the visualizations can be fed back to the algorithms by modifying, for example, parameters, or fed back to the data module in order to filter data or apply a different pre-processing technique.

1.3. Relation to other Work Packages

This deliverable D4.1, “Interactive visual analytics model”, primarily covers task T4.1, “Visual analytics model”, and already gives an outlook on task T4.2, “Feature exploration and parameter tuning”. Furthermore, the concepts and techniques described by the Visual Analytics Model constitute the basis for Tasks T4.3, “Visual analytics over spatial and temporal data”, and T4.4, “Visual analytics over network data”. It, therefore, serves as the theoretical foundation of the SDL-Vis component of SmartDataLake.

Due to Visual Analytics being of relevance in most of the components of SmartDataLake, WP4 in general and this deliverable in particular have a strong connection to tasks outside of WP4. In the following, the most related tasks are described.

WP1, Requirements, Architecture and Integration

- T1.1, “Use cases, requirements and integration” is already finished and defines requirements and scenarios that the Visual Analytics Model has to cover, including the final use-cases of the pilot partners.
- T1.2, “Design of system architecture” is already finished and explains the interplay between all components of SmartDataLake. The interactions defined in the Visual Analytics Model reflect the system architecture as characterized in this task.

WP3, Heterogeneous Information Network Mining

- T3.1, “Similarity search and exploration” and T3.2, “Entity resolution and ranking” will be finished simultaneously with T4.1. For the description of the Visual Analytics Model, they serve as concrete examples on how Visual Analytics will be implemented for task-driven knowledge generation and parameter tuning.
- T3.3, “Link prediction”, T3.4, “Community detection”, and T3.5, “Change detection and evolution” are upcoming tasks that will be implemented analogously to T3.1 and T3.2, following the definitions and concepts defined by the Visual Analytics Model.

2. The Visual Analytics Model

The Visual Analytics Model defines the pathways of interaction between the analyst and the Scalable and Interactive Visual Analytics Engine of SmartDataLake. It specifies how the analyst is included in the iterative visual analytics process, building the baseline for the interplay between automated machinery and human sensemaking. By characterizing the applied interfaces, visualizations and interaction techniques, the Visual Analytics Model describes how the analyst is supported in the explorative data analysis of the various mining tasks involved in SmartDataLake. For our project, we adopt the general Visual Analytics Model and combine it with the different analysis layers, as elaborated in the following.

2.1. The Visual Analytics Process

Interaction between human and machine is essential for data exploration and knowledge generation on large-scale data. While automated algorithms can process huge amounts of data and generate powerful models, knowledge generation not only requires results visualization but also tight human involvement in the Visual Analytics Process itself [2]. Figure 2 shows how the components of SmartDataLake relate with the general visual analytics model and how they interact with each other, and with the user to enable visual data analysis and sense-making.

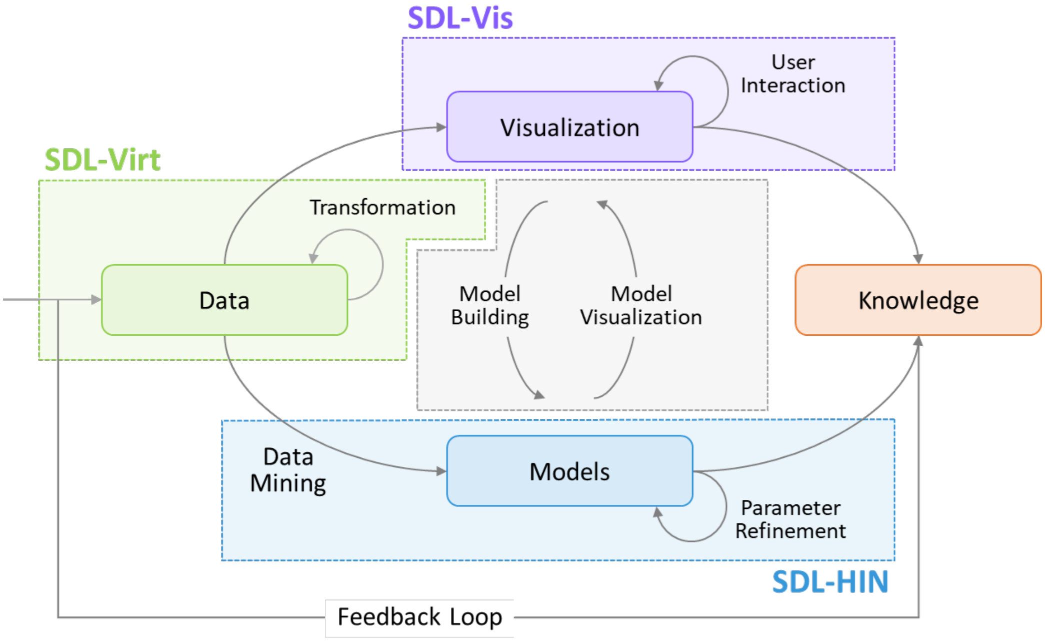


Figure 2: Embedding of the SmartDataLake components in the Visual Analytics Process by Keim et al. [2]. Knowledge generation is based on interactions between data, models, visualizations and the user.

2.1.1. VA Data Component: SDL-Virt

The *data component* of the visual analytics model is primarily covered by **SDL-Virt module**. This module provides unified access to heterogeneous and unstructured data in the data lake with the use of an SQL-like language for data filtering, transformation, and pre-processing. It enables access to *individual records*, as well as (both exact and approximate) *aggregate results*. The data type can have various *forms* (ranging from text, tabular, geo, network, and time-series data) and stored in different *locations*. All datasets need to be unified and merged in order to successfully analyze them.

Analysts need to understand the general structure of the data as well as the quality in order to perform a successful analysis. Examples of these analyses comprise showing statistics over valid entries in a dataset, calculating descriptive statistics, such as mean, minimum or maximum values, or the visualization of data distributions.

Before starting an analysis, the data need to be merged, cleaned, and pre-processed. In SmartDataLake this is especially difficult as the data lakes involve various data types which can already be structured, or without any structure at all.

The data acquired by SDL-Virt will be fed into the visualization component (SDL-Vis) and the machine learning module (SDL-HIN). Hence there is a tight coupling between the different modules.

SDL-Vis will support analysts by providing a multitude of tools to inspect the data. Standard tasks will be covered with state-of-the-art tools and commercial software, such as R & ggplot2, Python

& Jupyter, Tableau. However, more advanced analysis usually differs strongly for each use case and custom analysis tools need to be developed. These analyses comprise, for example, the general use-cases and tasks of the project (see Section 2.3.2, "Visualization and Interaction Panels").

Our developed tools will give analysts easy access to the data with the help of interactive systems. We will particularly focus on the inspection of the structure of data sources, the assessment of their quality and the analysis of data distributions. Understanding data distributions and their differences is a core feature in order to select and adjust models and visualizations of the other components.

2.1.2. VA Model Component: SDL-HIN

The *model component* of the visual analytics model is primarily covered by **SDL-HIN**. This module provides functionalities for model generation for various data assets. In particular, it supports similarity search over entity profiles, entity resolution and ranking, link prediction between entities, detection of entity groups and communities, and management of changes. Parameters of the different models can be refined by users (e.g., adjust similarity function or underlying clustering algorithms.

SDL-HIN is responsible for the model building process and typically involves a huge number of parameters which are often of very abstract nature. To define and refine parameters towards analysis goal and sensemaking analysts and intermediate analysis results need to be taken into account during the model building process. Furthermore, the analysis results need to be represented in a useful way so that analysts can understand the results and further modify the properties of a model, leading to better analysis results.

SDL-VIS will support analysts by providing meaningful visualizations of results and parameters. Hereby, we will focus on giving full insights and control over the important parameter choices and offering user-guidance for parameter choices. This indicates, for example, how parameter changes influence the mining results and thereby making the influence of the parameters transparent to the analyst. Hence, this enables a trustworthy knowledge generation and decision-making process.

The result visualizations will be newly developed and customized representations, particularly designed for the specific use-cases in order to give analysts the best possible insights.

2.1.3. VA Visualization Component: SDL-Vis

The *visualization component* of the visual analytics model is primarily covered by **SDL-Vis**. This module provides specialized visualization and interaction functionalities to represent the heterogeneous data assets in SDL, which include graph, spatial, and temporal visualizations. All visualizations can be interactively explored by analysts.

This deliverable describes the Visual Analytics Model, which builds the theoretical foundation for SDL-Vis. Therefore, SDL-Vis can be seen as the instantiation of the entirety of components and interfaces described in this document.

2.1.4. Relationship between Data, Model, and Visualization

SDL-Virt, SDL-Vis, and SDL-HIN are highly connected through the visual analytics model. The data can be fed into the visualization and model layer. Different representations can be used to get an overview of the data, and data mining algorithms can be performed to identify patterns in the data. Both can be led to new insights and knowledge about the data with respect to a concrete analysis task. This knowledge can then be injected back into the data module through a feedback loop in order to refine data selection, transformation, filtering, and data pre-processing.

There is also a close relationship between SDL-Vis and SDL-HIN. The machine learning models derived by SDL-HIN can be visualized in order to understand the general structure of the model and the results after applying the model (Model Visualization). Findings derived by an exploratory data analysis using SDL-Vis can be injected into the model component in order to select and refine parameters and choose appropriate data mining and machine learning algorithms.

Ultimately, this inter-relationships help with the *model building* and *model visualization* process; which is of an iterative nature; also supported by the domain knowledge of the analysts: the user learns from the machine through the visualizations, and, vice versa, the machine learns from the human which is injected by modified parameters, and selected algorithms.

2.1.5. Relationships from a more practical view

To enable the user to interact with all relevant components in the data analysis workflow, SDL-Vis will provide interfaces for both data (SDL-Virt) and models (SDL-HIN). For the heterogeneous data sources of SmartDataLake, data can have a variety of different forms, making data profiling an essential task of the visual analytics process. Therefore, SDL-Vis will provide descriptive analytics in the form of statistical information about the data sources. For the data mining and model building algorithms of SDL-HIN, the right choice of parameters is fundamental. While this can initially be done semi-automatically, user interaction and refinement are crucial to adapt the parameters according to the knowledge the user gains during the analysis process. This, in return, influences the model, resulting in an iterative model building, human sense-making, and model refinement process.

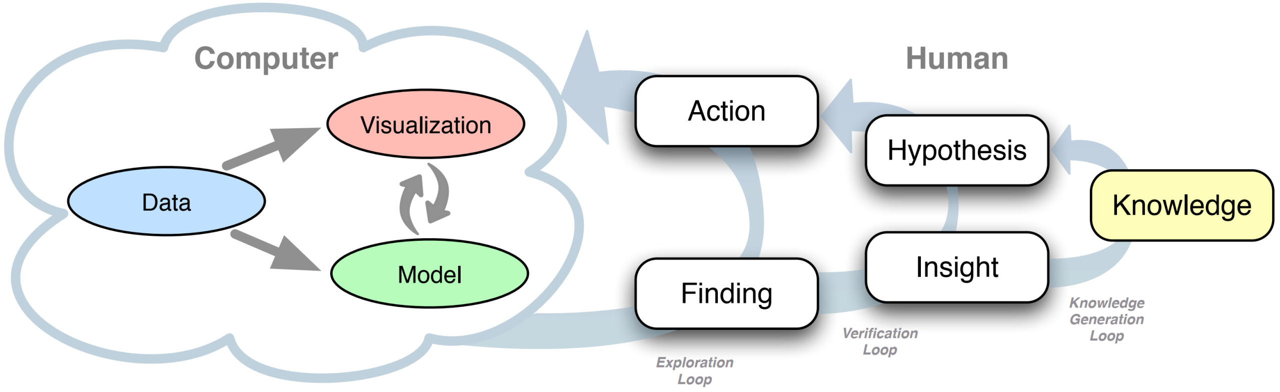


Figure 3: Knowledge generation model for visual analytics [3]. Extending the Visual Analytics Process by Keim et al. [2], it depicts how human and computer complement each other in the data exploration, hypothesis verification and knowledge generation process.

Sacha et al. [2] extend the Visual Analytics Process including this iterative human-machine interplay to the Knowledge Generation Model for Visual Analytics, as shown in Figure 3. Here, the process of *knowledge generation* is broken down into data exploration, hypothesis verification, and knowledge generation. Particularly relevant for SmartDataLake is the exploration loop. The heterogeneous data sources require an extensive pre-processing and data mining setup, where individual decisions and parameters have to be explored during analysis. Human creativity is essential to generate new *findings*, leading to required *actions* for changes in the pipeline. This affects all three components of the automated part, namely *data*, *model*, and *visualization*. The design of SDL-Vis is tailored to represent these three components.

2.2. SDL-Vis Architecture

The architecture of SDL-Vis is a fundamental part of the Visual Analytics Model, describing its interfaces and interactions with each of the lower-level components of SmartDataLake. It builds the basis for the development of SDL-Vis and its integration in SmartDataLake according to the Knowledge Generation Model for Visual Analytics [3].

Being the primary interface for accessing the functionality of SmartDataLake, SDL-Vis covers two major aspects: (1) enabling human sense-making of results through providing meaningful visualizations, and (2) allowing interaction with these results to adjust the parameterization of underlying algorithms. Visualization and interactions are handled by the Visual Explorer component of SDL-Vis, which, therefore, provides the user-frontend. Possibly necessary data pre-processing, aggregation, or transformation of results of the lower-level components of SmartDataLake is handled by the Visual Analytics Engine. It serves as the backend of SDL-Vis, exposing its functionality through a REST-API. Figure 4 shows a diagram of the SDL-Vis architecture.

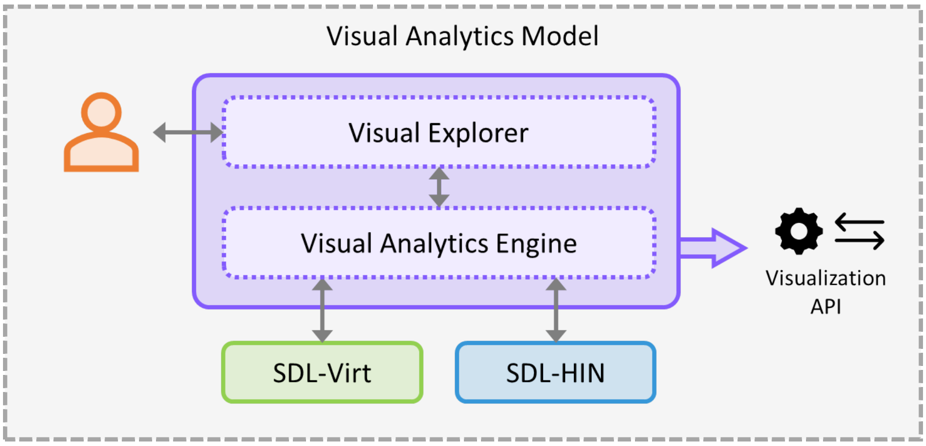


Figure 4: Architecture of SDL-Vis.

While the Visual Analytics Engine is implemented as a standalone server application, the Visual Explorer is realized as a Web Frontend to keep maximum flexibility and scalability.

2.3. User Interfaces (Visual Explorer)

The user interface of SDL-Vis, namely *Visual Explorer*, will combine multiple interactive visualization components in the form of *visualization and interaction panels*. Each panel covers a specific analysis task, providing visualizations on the results and parameters of the lower-level components of SmartDataLake.

Like the name implies, Visual Explorer especially targets the exploration loop of the knowledge generation model. The information hidden in large, heterogeneous data collections, such as SmartDataLake handles, is virtually never known a priori and, therefore, has to be iteratively discovered during the analysis process. To support human actions regarding automated parts of SmartDataLake during the exploration loop, parameters of the analysis pipeline are visualized and can be adjusted to modify results towards the user's insights and finding (see Figure 5). Such *actions* are supported by directly linking interaction techniques to the visualizations.

Human actions are relevant during all steps of the knowledge generation model. Human actions generally trigger system reactions. Observation and inspection of these reactions, in return, lead to findings by the analyst. In the following, the relevant types of actions and respective entities they are applied to are explained, building the foundation for the structural and visual design of Visual Explorer.

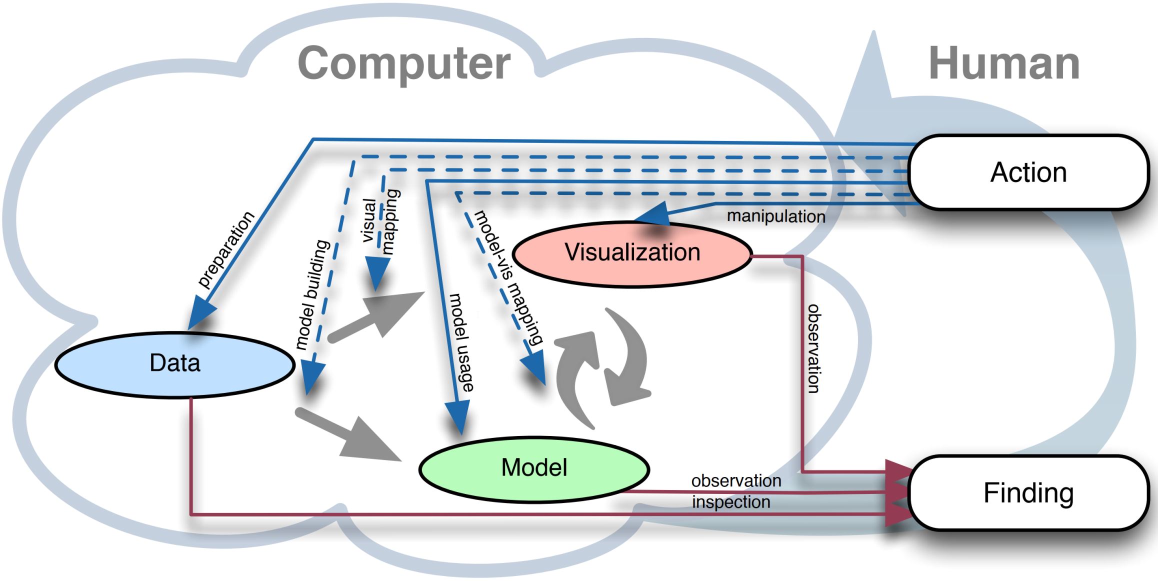


Figure 5: Action and cognition paths in the VA process [3].

2.3.1. Main Actions

2.3.1.1. Preparation

The *preparation action* is applied to the data entity. It describes the collection, selection and consolidation of data. For SmartDataLake, this means to integrate different heterogeneous data sources of varying type and quality. By *inspecting* the system's reaction to the data preparation action, the user gets an understanding on the structure and peculiarities of the available data and the necessary pre-processing steps. This action-reaction loop is covered by SmartDataLake as the *data profiling* task.

2.3.1.2. Model Building / Model Usage

The *model building action* relates to the refinement of the knowledge discovery process and the data mining pipeline. Building meaningful models from the data is a fundamental goal of the Visual Analytics process. Applying the models to actual data is referred to as the *model usage action*, leading to observable results. Making use of these observations, the user updates his understanding of models and results. The VA pipeline of SmartDataLake involves multiple, use-case driven models (e.g., similarity search, entity ranking, entity resolution, and community detection) that require an advanced parameterization. Therefore, the model building and model usage actions are a central part of SmartDataLake, covered in the form of the *parameter tuning* task.

2.3.1.3. Visual Mapping / Model-Vis Mapping

Visual mappings are a core part of Visual Analytics. This covers both actions, the *visual mapping* of data, as well as the *visual mapping of models*. Since data and models are typically complex and require a powerful abstraction, visualizations are essential for communication and understanding. Viewing and interpreting the visualizations is referred to as observation, which generates new findings about data and knowledge discovery process, finally leading to new actions based on the new understanding and updated expectations of the analyst. The multitude of models and the complexity of data involved in the SmartDataLake VA process requires targeted visual mappings for individual sub-tasks. Therefore, the Visual Explorer user interface contains distinct views, linked to the respective tasks and analysis goals. The actions are implemented in SDL-Vis in the form of *results visualizations* and *parameter tuning* visualizations.

2.3.1.4. Manipulation

Actions do not inherently result in an update of data or models but can also only affect the visualization itself. This action is called *manipulation* and describes the interaction with the visualization. For example, the user might pan and zoom to parts of interest, filter, highlight or save results. This helps the user in understanding the visual mappings of models and data, which might indirectly result in new actions applied to those entities. The manipulation action is highly task-driven and, therefore, is implemented in different forms in Visual Explorer.

2.3.2. Visualization and Interaction Panels

Finding a good representation of results in an analysis process is highly dependent on the data, the domain and the goal of the analysis. With increasing complexity of the data and the information hidden inside, the visualization and interaction methods need to be highly customized. In Visual Analytics, virtually all results are based on an automated analysis pipeline representing the models that are refined during the exploration and sensemaking process [3]. Understanding the process of how results were generated is fundamental for a full understanding of the results themselves. Therefore, Visual Analytics has to communicate the models along with the results. This often is done in the form of visualizing the parameters that have the largest influence on the quality of the results. For example, these could be thresholds for clustering methods (e.g., k for k-Means [11]), configurations for iterative projection methods (e.g., perplexity, iterations, learning rate, and momentum for t-SNE [12]), or parameters of newly created algorithms (e.g., weights for SmartDataLake similarity search algorithms). Understanding these parameters is not only essential for a meaningful interpretation of the produced results, but also a necessity for the model building action (see section 2.3).

The advanced tasks Visual Analytics has to solve requires highly specific user interfaces, visualizations and interaction possibilities. To allow the analyst to focus on relevant details, information overload has to be avoided as much as possible. Two major strategies are available to prevent information overload:

1. automatically supporting the user in the analysis task, or

2. splitting an overarching task into semantically independent sub-tasks.

If the analysis goal is complex and dependent on a powerful user interface supporting many options, a task split might not be possible. In that case, supporting and guiding the user during the analysis is essential. For example, this can be done by automatically filtering or highlighting information, indicating the expected outcome for a future action, or by even proposing possibly useful next steps to the user [8].

If a complex task can be further split into smaller analysis tasks, distinct user interfaces and customized visualizations help to avoid visual clutter and allow the analyst to focus on relevant information. To prevent the separation of naturally related real-world data and, thereby, the loss of possibly important information, techniques and visual encodings for context preservation, such as linking [13] and brushing [14], have to be implemented.

For the complexity of data and mining algorithms involved in SmartDataLake, SDL-Vis has to cover various tasks and analysis goals, some even occurring at different stages of the analysis pipeline. This makes the distinct tasks semantically separable and allows SDL-Vis to initially simplify the analysis process by providing tailored user interfaces, visualizations and interaction techniques for each of them.

Therefore, Visual Explorer will provide different analysis panels, each of them tailored to a specific analysis task of SmartDataLake. The tasks are covering either a single or multiple use cases, as defined in deliverable D1.1, "Use Cases and Requirements" of SmartDataLake. In the following, the planned views are described in conjunction with the respective tasks and use cases they will be applied to. This includes visualization design, displayed data, and input/output parameters for each view, covering observations and actions from and towards automated parts of the analysis pipeline.

2.3.2.1. Data Profiling Panel

Before starting data mining and data analysis tasks, data profiling is a mandatory step for real-world datasets in data lakes. They often originate from heterogeneous data sources, are of diverse structure, and contain noise or missing values. Therefore, the data scientist has to closely inspect new data to get familiar with its quality and structure [15].

Since SmartDataLake involves highly heterogeneous data types and sources, data profiling is an essential task in the data processing and sensemaking pipeline. Use-case A.2, "Computing Descriptive Analytics" of deliverable D1.1, "Use Cases and Requirements" describes the needs on the data profiling task of the SmartDataLake pilot partners. To inspect the data and get a first impression of its structure and quality, *descriptive analytics* is fundamental. Due to the diversity and complexity of the datasets to be analyzed, the pilot partners rely on a customized workflow. While the required functionalities for data pre-processing and the calculation and visualization of descriptive statistics can be solved with pre-existing libraries, the workflow itself changes from case to case. To provide such flexible environment with a powerful connection to existing statistical libraries and visualization tools, *Jupyter notebooks* [16] are used for data profiling in SmartDataLake.

For basic data profiling tasks (e.g., descriptive statistics, charts), existing solutions for Business Intelligence (e.g., Tableau[1], Qlik[2], Microsoft Power Bi[3], SAP BusinessObjects Business Intelligence[4]) could be utilized. However, those tools reach their limits for the inspection of the structure of the data, such as available attributes or data domain. Conversely, these aspects have to be considered prior to importing the data in BI applications, since they rely on a pre-processed and structured form of data as input. Therefore, SmartDataLake relies on a more flexible solution based on well-suited programming languages for data processing and visualization.

While other programming languages and packages are available to cover the task of data profiling (see Table 1), SDL-Vis relies on the combination of Python and Jupyter. Jupyter notebooks have gained popularity in the last few years for scientific computing and publishing, since they transparently show source code side-by-side with generated results. This ensures reproducibility, allows easy modification and sharing, and integrates well with version control systems like git[5]. Furthermore, the variety of easy-to-use packages for mathematical computing and visualization makes Python an ideal choice for rapid prototyping of data profiling applications.

Language [Environment]	Packages
R[6] [Jupyter[7]]	ggplot2[8]
Python[9] [Jupyter]	**NumPy[10], pandas[11], SciPy[12], matplotlib[13]**
HTML / JavaScript	Vega/Vega-Lite[14], D3[15]

Table 1: Common programming languages and corresponding packages for data processing, descriptive statistics and data visualization.

Since the available tools perfectly cover the requirements of the SmartDataLake data profiling use cases, SDL-Vis integrates a JupyterLab environment in Visual Explorer. This keeps full flexibility while at the same time allowing for a fast context-switch between data profiling and analysis panels (see Figure 6).

Use-case driven data profiling scripts can be directly edited, saved, and shared from the integrated JupyterLab UI. The pilot partners being used to perform customized data inspection with Python/Jupyter ensures an easy shift to the SmartDataLake environment, as well as the preservation of existing workflows.

[1] https://www.tableau.com/

[2] https://www.qlik.com/

[3] https://powerbi.microsoft.com/

[4] https://www.sap.com/products/bi-platform.html

[5] https://git-scm.com/

[6] https://www.r-project.org/

[7] https://jupyter.org/

[8] https://ggplot2.tidyverse.org/

[9] https://www.python.org/

[10] https://numpy.org/

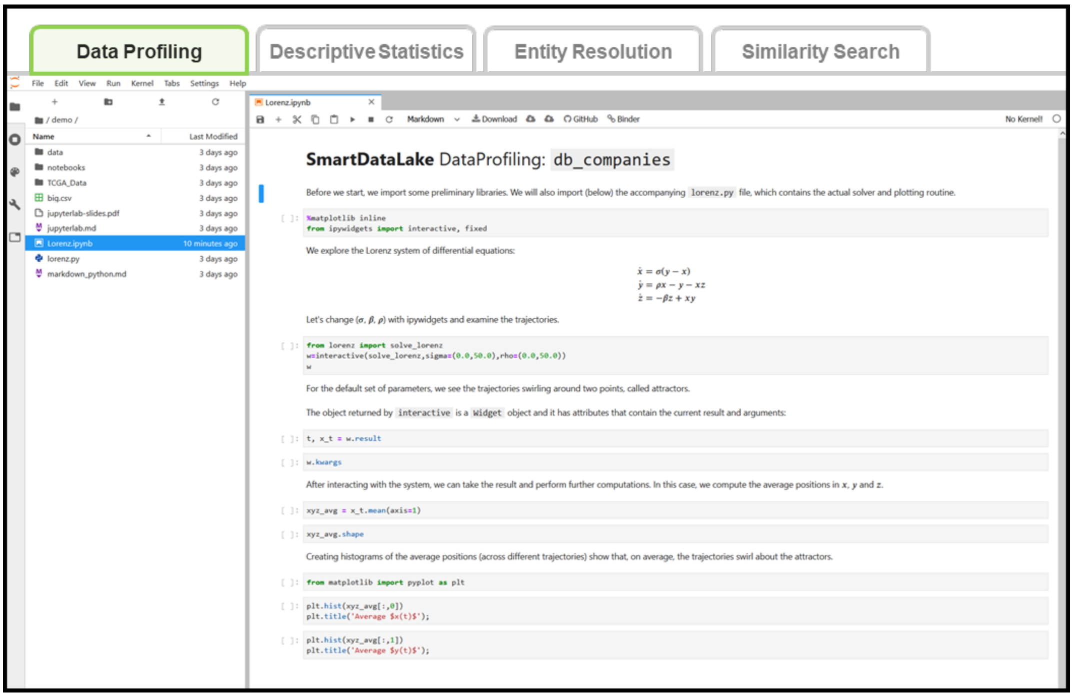
[11] https://pandas.pydata.org/

[12] https://www.scipy.org/scipylib/index.html

[13] https://matplotlib.org/

[14] https://vega.github.io/

[15] https://d3js.org/



Name	Last Modified
data	3 days ago
notebooks	3 days ago
TCGA_Data	3 days ago
biq.csv	3 days ago
jupyterlab-slides.pdf	3 days ago
jupyterlab.md	3 days ago
Lorenz.ipynb	10 minutes ago
lorenz.py	3 days ago
markdown_python.md	3 days ago

SmartDataLake DataProfiling: `db_companies`

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

```
[ ]: %matplotlib inline
     from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= \rho x - y - xz\\ \dot{z} &= -\beta z + xy\end{aligned}$$

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
[ ]: from lorenz import solve_lorenz
     w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
     w
```

For the default set of parameters, we see the trajectories swirling around two points, called attractors.

The object returned by `interactive` is a `Widget` object and it has attributes that contain the current result and arguments:

```
[ ]: t, x_t = w.result
```

```
[ ]: w.kwargs
```

After interacting with the system, we can take the result and perform further computations. In this case, we compute the average positions in x, y and z.

```
[ ]: xyz_avg = x_t.mean(axis=1)
```

```
[ ]: xyz_avg.shape
```

Creating histograms of the average positions (across different trajectories) show that, on average, the trajectories swirl about the attractors.

```
[ ]: from matplotlib import pyplot as plt
```

```
[ ]: plt.hist(xyz_avg[:,0])
     plt.title('Average $x(t)$');
```

```
[ ]: plt.hist(xyz_avg[:,1])
     plt.title('Average $y(t)$');
```

Figure 6: Data profiling panel. An integration of a Python/Jupyter environment keeps flexibility while a common interface allows for fast context-switch between data profiling and analysis panels.

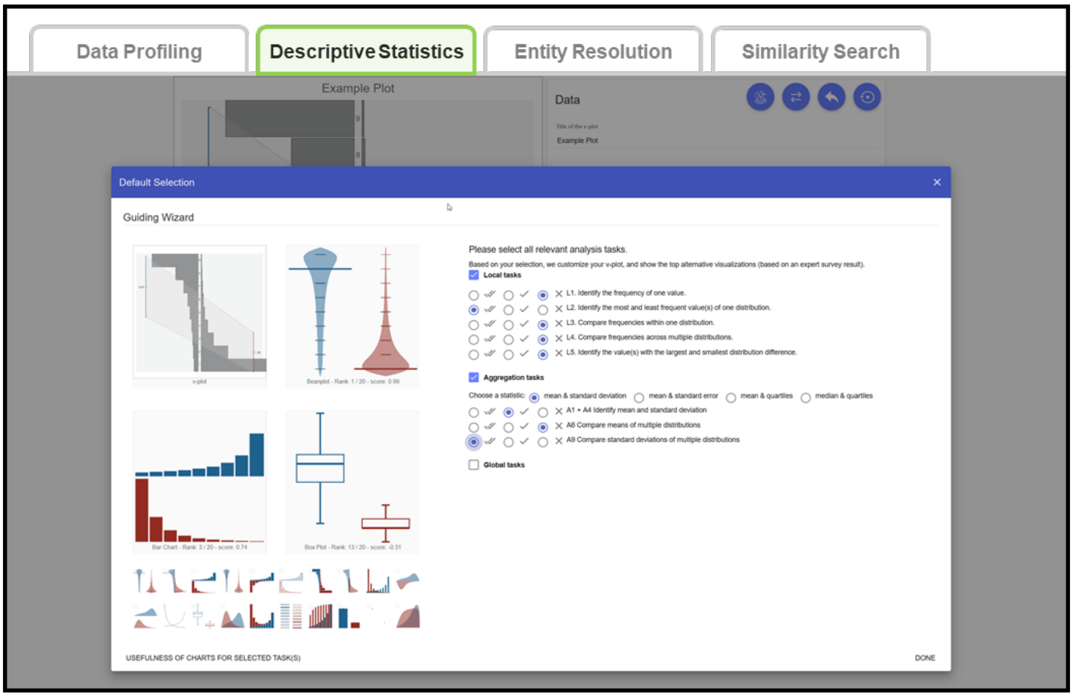
2.3.2.2. Descriptive Statistics Panel

Besides the customized workflow of the data profiling panel, user guidance is an important aspect during analysis of new data. After the general structure of the data foundation is understood, higher-level statistics about the data become important. As defined in use case A.2, "Computing Descriptive Analytics" of deliverable D1.1, "Use Cases and Requirements" the user needs the ability to compute statistical descriptors (e.g., average, minimum, maximum) over the available data. While this could be done manually using the data profiling panel, user guidance might be important to choose the ideal visual representation regarding the specific task and analysis goal.

Complementing the functionality of the data profiling panel, the *descriptive statistics panel* includes a Visual Analytics frontend that supports the user in finding task-oriented visualizations for descriptive statistics. Depending on the analysis tasks, different visualizations might be best-suited. For example, while boxplots [1] clearly communicate median and spread of a value distribution, they fail when a comparison of individual values is required. Hybrid charts tackle the problem of finding an ideal visualization for a given task by combining multiple individual charts to a task-specific ensemble. However, this is a time-consuming process, often requires programming skills and presumes detailed knowledge on the suitability of a visualization for a specific task.

To tackle this issue, with the *v-plots designer*[16], we propose a system which supports and guides the user in creating a hybrid chart based on dataset and analysis goal [17]. The v-plot is a layered representation, combining five (*Rom.*: "V") different chart types where each of them supports a different analysis task. To evaluate which type of chart is best-suited for a certain task, we performed an extensive user-study with domain experts. From the study results, guidelines were derived which are used as the foundation for an automated guiding wizard. The wizard guides the user in optimizing a v-plot according to the analysis task.

While the data profiling panel offers a highly customized analysis workflow, v-plots enables a fast and reliable inspection of data distributions. Therefore, Visual Explorer will integrate the v-plots designer in an additional *Descriptive Statistics Panel*, as shown in 2.3.2.2. The automated configuration wizard allows to find best-suited charts for the analysis of the distributions in a dataset and, therefore, provides otherwise possibly disregarded insights. Based on the analysis tasks of the user, the v-plot designer automatically proposes appropriate visualizations and, furthermore, builds a customized v-plot that combines all relevant tasks in one hybrid chart.



Please select all relevant analysis tasks.

Based on your selection, we customize your v-plot, and show the top alternative visualizations (based on an expert survey result).

☑ **Local tasks**

○ ✓✓ ○ ✓ ● ✗ L1. Identify the frequency of one value.
● ✓✓ ○ ✓ ○ ✗ L2. Identify the most and least frequent value(s) of one distribution.
○ ✓✓ ○ ✓ ● ✗ L3. Compare frequencies within one distribution.
○ ✓✓ ○ ✓ ● ✗ L4. Compare frequencies across multiple distributions.
○ ✓✓ ○ ✓ ● ✗ L5. Identify the value(s) with the largest and smallest distribution difference.

☑ **Aggregation tasks**

Choose a statistic: ● mean & standard deviation ○ mean & standard error ○ mean & quartiles ○ median & quartiles

○ ✓✓ ● ✓ ○ ✗ A1 + A4 Identify mean and standard deviation
○ ✓✓ ○ ✓ ● ✗ A6 Compare means of multiple distributions
● ✓✓ ○ ✓ ○ ✗ A9 Compare standard deviations of multiple distributions

☐ **Global tasks**

Figure 7: Descriptive statistics panel. Using V-Plot [17], the analyst is supported in building charts that are tailored to data and analysis task.

2.3.2.3. Entity Resolution Panel

SmartDataLake integrates information from various, heterogeneous data sources. Merging these sources is a complex task, since data might be of different quality and come in structured, semi-

[16] https://v-plot.dbvis.de

structured or unstructured form. Furthermore, even for structured or semi-structured data, attributes or tags might differ across multiple data sources. For further analysis, data has to be merged automatically across all available sources in a meaningful way. A fundamental problem is *entity resolution*, i.e., combining data entities which represent the same real-world entity. This requirement is described in use case A.3, "Matching company profiles across sources" and use case C.1, "Resolving entity profiles across sources" of deliverable D1.1, "Use Cases and Requirements" of SmartDataLake.

The data of SmartDataLake can be viewed logically as a graph, in particular a *heterogeneous information network* (HIN). The graph contains nodes for each data entity, with links representing the relation between these entities. Both nodes and links may have attributes attached to them (e.g., the type of link relation), building a *property graph*. Speaking in terms of the HIN property graph of SmartDataLake, entity resolution can be re-formulated as the problem to insert *equality*-relations between data entities that represent the same real-world entity, based on specific properties of these entities. Since entity resolution is part of the automated analysis pipeline of SmartDataLake, a model has to be built which defines when and how entities should be joined (i.e., *entity resolution threshold*). Building this model is not trivial and depends on multiple parameters that have to be interactively explored during the analysis.

The *Entity Resolution Panel* is based on a hierarchical visualization of the HIN property graph, which will be a central part of the upcoming deliverable D4.4, "Visual Analytics over Network Data". It allows the inspection of multiple, similar entities and how they were merged. The parameters responsible for the currently displayed results are visualized next to the graph. By examining the introduced links between entities, as well as their attributes and data sources, the analyst can decide if the correct entities were merged by the entity resolution algorithm. Insights can directly be translated into changes in the parameter set, functioning as action for a model update (compare to section 2.3). The change in the model state leads to an updated result set. Changes are directly highlighted in the HIN graph, allowing the analyst to observe the influence of his changes. This allows an iterative tuning of parameters, helping the analyst understanding the logic of the entity resolution engine. Besides modifying the parameters, the user can directly interact with the graph representation. Tools to split and merge distinct entities are translated into respective changes in the parameter set, that lead to the required changes.

Since the HIN graph of a full SmartDataLake instance potentially contains millions of nodes, manipulation of the visualization is a fundamental functionality of the entity resolution panel. The hierarchical representation of the graph allows a significant reduction of visible entities. This is not only relevant to avoid information overload, but also affects computation time, enabling real time analysis. Figure 8 shows the visualization of a leaf node of the hierarchy, merged from multiple data entities. Changes observed after a parameter update are indicated in red.

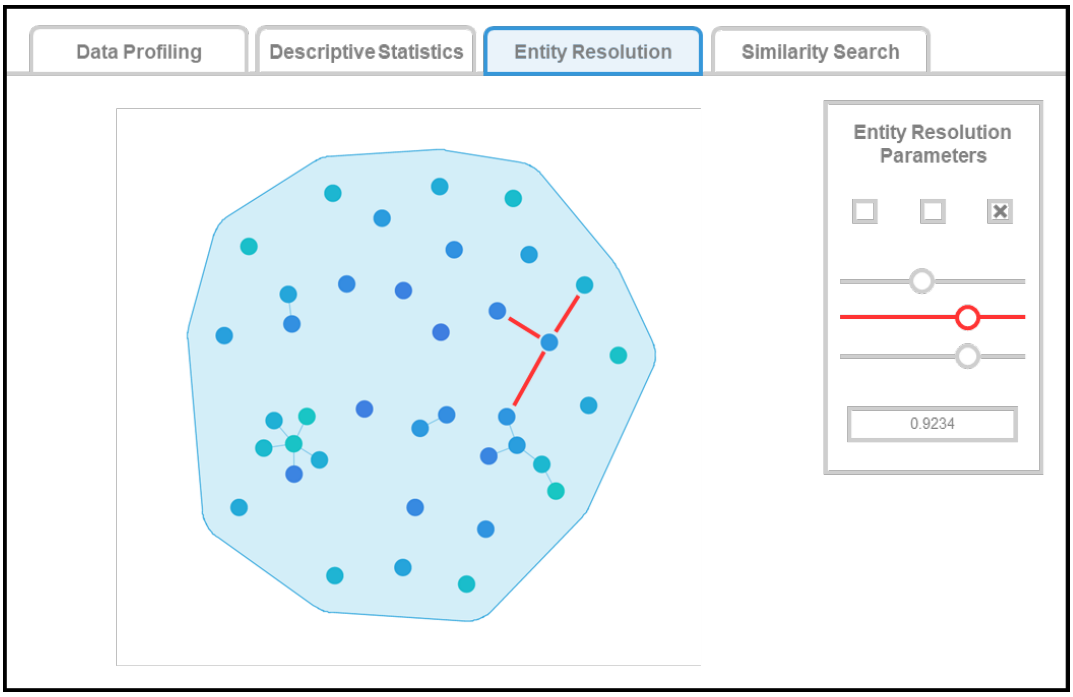


Figure 8: Entity resolution panel. The HIN property graph is visualized in the form of a hierarchy. Entity resolution parameters can either be updated using the value display on the right, or by directly interacting with the links and nodes of the graph.

2.3.2.4. Similarity Search Panel

In large data collections, search is the most powerful way to identify entities of interest. However, the exact attribute values for the desired entities are often not known a priori, or more than one entity might be of interest. For example, if a company wants to compare itself with competitors of similar size and location, the exact number of employees or the location is not known beforehand. Therefore, *similarity search* enables the user to identify entities that are close to the desired combination of attributes. This requirement is defined in use-case A.4, “Finding similar entity profiles” of deliverable D1.1, “Use Cases and Requirements” of SmartDataLake.

Since there typically is no entity exactly matching the search parameters, not only the most similar entity has to be considered. Instead, a further analysis of the top-k results is necessary to find those entities that are of interest to the analyst according to the current analysis task. For each entity in the result set, a *similarity score* is calculated, which is used to rank the results. More similar results have a higher similarity score, resulting in a higher rank of those results.

The search query can consist of multiple search criteria of possibly different data domains. For example, a search by keyword involves categorial data, a search by numbers involves numerical data, and a search by geolocation involves spatial data. This leads to the problem of weighting similarities against each other (see Figure 9): is a company with a spatial distance of 50km and 20 employees more similar to a company with 10 employees than a company with a distance of 10km and 70 employees? The similarity search algorithm of SmartDataLake, as described in deliverable

D3.1, "Similarity search, entity resolution and ranking", solves this issue by multiplying the similarity score of each search criterion by a user-specified scalar value. This way the analyst can steer the entity exploration according to his domain knowledge (see Figure 10).

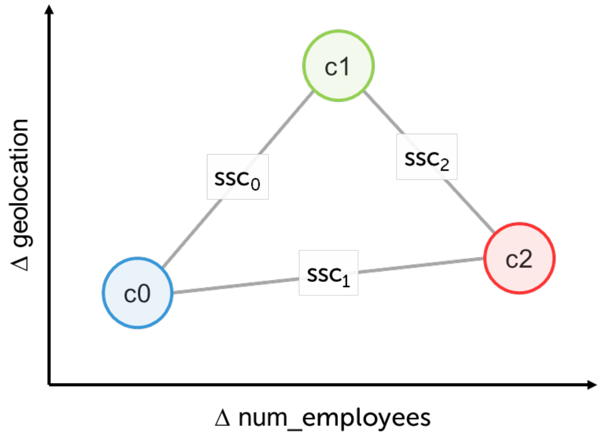


Figure 9: Similarity search results for different data domains. The similarity scores can not be directly compared, since they have different meanings

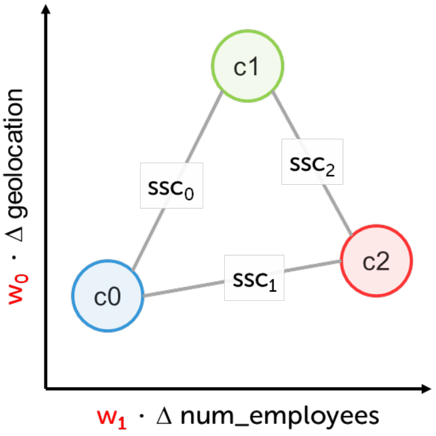
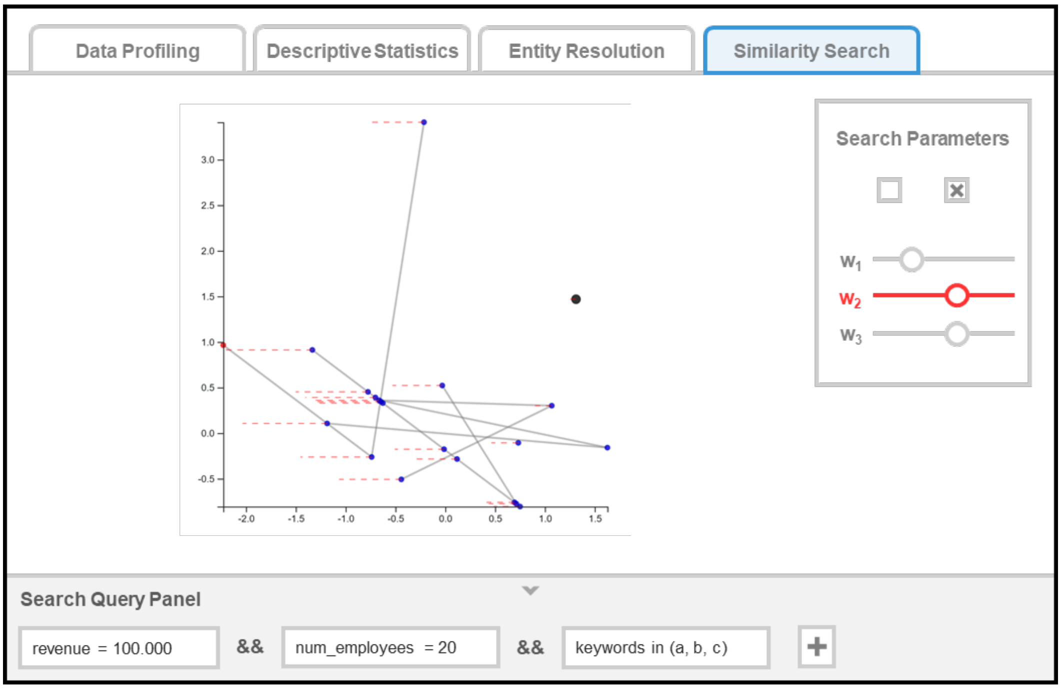


Figure 10: By applying weightings to the similarity scores, the analyst is able to steer the results according to his knowledge.

While the weightings give the analyst full control over the influence of each search parameter, it introduces additional complexity. The weights are rather abstract parameters, for which the choice of meaningful values is not obvious by default.

To solve this issue, the *similarity search panel* of Visual Explorer supports the analyst in understanding and refining these abstract parameters in an interactive exploration loop. This is done by combining the visualization of results with feedback on past as well as future decisions. The visualization shows a projection of the top-k results that are returned for a search query with search parameters p_0, ..., p_n with a fixed set of weights w_0, ..., w_n. The projection indicates pairwise similarities between entities of the result set as edges between the projected points. Furthermore, the root search query, i.e., the parameters that were searched for, is displayed as a virtual point in the result set indicating a perfect match.

When the user shows the intention to update the weight configuration, the system gives feedback on the expected outcome of this change, combining the strengths of human sensemaking and computational resources. On-the-fly computation of various possible parameter combinations enables real-time interaction with the system and gives the analyst a first impression if a choice is useful to reach the analysis goal. Changes are visually encoded by updating the projection with the new result set, while simultaneously showing how existing points move after changing weights. Newly points in the result set are highlighted, while vanishing points are faded out. Figure 11 shows the similarity panel, indicating the updates in the result set if the user would apply a certain change to weight w_2.



revenue = 100.000 && num_employees = 20 && keywords in (a, b, c)

Figure 11: The data profiling panel of Visual Explorer. An integration of a Python/Jupyter environment keeps flexibility while a common interface allows for fast context-switch between data profiling and analysis panels.

While projections work well for numerical data, and reasonably well for categorial data (e.g., by applying one-hot encodings), spatial data cannot directly be embedded in this form of visualization. The similarity search panel, therefore, will also have an option to switch to a map-based representation, e.g., by warping the map according to the projection of non-spatial attributes. Together with displaying the influence of parameter and weight updates over time, this will be part of task T4.3, "Visual Analytics over Spatial and Temporal Data".

2.4. Backend (Visual Analytics Engine)

Virtually every form of visual data representation needs pre-processing. Limitations in display space, human cognition and retentiveness, and computational complexity make abstraction inevitable. Moreover, the representation of data has massive influence on how information is interpreted by humans. Visualization research, therefore, is about finding a *meaningful* abstraction, affording pre-processing, transformation and aggregation of data.

Since this is typically a memory-intensive and computationally expensive process, it should be avoided to be done on client-side. Furthermore, dependent on the complexity of the data processing pipeline, intermediate results of automated components may have to be buffered, cached or stored for later use. To avoid redundancies in the computation of such results, this has to be done at a central point.

All these requirements can be fulfilled by partitioning SDL-Vis into two applications: frontend and backend. The frontend, Visual Explorer, is responsible for the presentation of results, while the backend, the Visual Analytics Engine, handles complex data processing, storage, and abstraction tasks. Results of the Visual Analytics Engine are exposed via REST API, ensuring a standardized and well-documented interface for the provided services and making both applications implementation-wise independent.

The Visual Analytics Engine directly interfaces with the lower-level components of SmartDataLake, sending requests to models with a certain set of parameters and receiving responses containing the respective results.

3. Conclusion

This report defines the Visual Analytics Model as the primary subject of deliverable D4.1, "Interactive Visual Analytics Model". It, thus, forms the basis for all upcoming tasks of WP4, "Scalable and Interactive Visual Analytics".

To make sense from the large-scale, heterogeneous data sources of SmartDataLake, Visual Analytics is an essential requirement. By incorporating human sensemaking tightly with automated parts of the analysis pipeline, understanding of both results and underlying models can be achieved. This builds the foundation for trustworthy decisions and enables meaningful adjustments of models according to the analyst's knowledge and analysis goals.

All components of SmartDataLake are affected by the Visual Analytics Model; it defines the interplay between automated parts of the analysis pipeline and human sensemaking. The Visual Analytics Model describes user interfaces, visualizations and interaction methods for the three major use-cases of WP4, namely data profiling, parameter optimization, and the visualization of results. It, therefore, builds the theoretical foundation for the implementation of SDL-Vis.

Since SmartDataLake involves highly specific tasks, the user interface of SDL-Vis, Visual Explorer, will contain multiple task-driven Visualization- and Interaction Panels. Concrete examples of these panels demonstrate how Visual Analytics is implemented in SDL-Vis. Each task has its own specific data and models involved, requiring a customized knowledge generation workflow. Finally, SDL-Vis combines all Visualization- and Interaction Panels in an overarching user interface, allowing easy context switch and portability of results.

For the transformation, aggregation and storage of intermediate results, SDL-Vis requires a backend application, the Visual Analytics Engine. It performs computationally and storage-wise expensive operations to take load from the client side and, thus, allow smooth exploration by the analyst.

With the Visual Analytics Model, this deliverable lays the foundation for the upcoming tasks of WP4. It serves as reference not only for the implementation of SDL-Vis, but also defines interfaces and interaction patterns for all other components of SmartDataLake that are involved in the iterative knowledge generation and sensemaking process.

Based on the tasks, methods and interfaces defined by the Visual Analytics Model, we are currently working on the implementation of functionalities for feature exploration and parameter tuning, which will be presented in deliverable D4.2.

References

[1] J. Tukey, Exploratory Data Analysis, Reading, Mass: Addison-Wesley Pub. Co, 1977.

[2] D. Keim, J. Kohlhammer, G. Ellis and F. Mansmann, Mastering The Information Age – Solving Problems with Visual Analytics, Eurographics Association, 2010.

[3] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis and D. A. Keim, “Knowledge Generation Model for Visual Analytics,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 20, pp. 1604-1613, 12 2014.

[4] G. U. Yule, “Why do we Sometimes get Nonsense-Correlations between Time-Series?–A Study in Sampling and the Nature of Time-Series,” *Journal of the Royal Statistical Society,* vol. 89, p. 1, 1 1926.

[5] Y. L. Simmhan, B. Plale and D. Gannon, “A Survey of Data Provenance Techniques,” 2005.

[6] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis and D. A. Keim, “The Role of Uncertainty, Awareness, and Trust in Visual Analytics,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 22, p. 240–249, 1 2016.

[7] D. Sacha, “Knowledge Generation in Visual Analytics : Integrating Human and Machine Intelligence for Exploration of Big Data,” Konstanz, 2018.

[8] F. Sperrle, J. Bernard, M. Sedlmair, D. Keim and M. El-Assady, “Speculative Execution for Guided Visual Analytics,” *Machine Learning from User Interactions for Visualization and Analytics: An IEEE VIS 2018 workshop,* 7 8 2019.

[9] Z. Abedjan, L. Golab and F. Naumann, “Profiling relational data: a survey,” *The VLDB Journal,* vol. 24, pp. 557-581, 6 2015.

[10] F. Nargesian, E. Zhu, R. J. Miller, K. Q. Pu and P. C. Arocena, “Data lake management,” *Proceedings of the VLDB Endowment,* vol. 12, p. 1986–1989, 8 2019.

[11] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Berkeley, Calif., 1967.

[12] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” 2008.

[13] A. Buja, J. A. McDonald, J. Michalak and W. Stuetzle, “Interactive data visualization using focusing and linking,” in *Proceeding Visualization \textquotesingle91*, 1991.

[14] R. A. Becker and W. S. Cleveland, “Brushing Scatterplots,” *Technometrics,* vol. 29, pp. 127-142, 1987.

[15] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, Elsevier LTD, Oxford, 2017.

[16] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing and J. development team, “Jupyter Notebooks - a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 2016.

[17] M. Blumenschein, L. J. Debbeler, N. C. Lages, B. Renner, D. A. Keim and M. El-Assady, “v-plots: Designing Hybrid Charts for the Comparative Analysis of Data Distributions,” *Computer Graphics Forum,* vol. 39, 2020.

[18] StackOverflow, *Developer Survey,* 2019.