

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,300

Open access books available

130,000

International authors and editors

155M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Implementing Visual Analytics Pipelines with Simulation Data

*Taimur Khan, Syed Samad Shakeel, Afzal Gul,
Hamza Masud and Achim Ebert*

Abstract

Visual analytics has been widely studied in the past decade both in academia and industry to improve data exploration, minimize the overall cost, and improve data analysis. In this chapter, we explore the idea of visual analytics in the context of simulation data. This would then provide us with the capability to not only explore our data visually but also to apply machine learning models in order to answer high-level questions with respect to scheduling, choosing optimal simulation parameters, finding correlations, etc. More specifically, we examine state-of-the-art tools to be able to perform these above-mentioned tasks. Further, to test and validate our methodology we followed the human-centered design process to build a prototype tool called ViDAS (**V**isual **D**ata **A**nalytics of **S**imulated **D**ata). Our preliminary evaluation study illustrates the intuitiveness and ease-of-use of our approach with regards to visual analysis of simulated data.

Keywords: visual analytics, machine learning, interaction, user experience, assistive technologies

1. Introduction

This section gives a brief overview of this chapter. It starts by discussing what usability is and why advanced analysis is required to extract useful information from raw data. Later, the idea and need for visual analytics is discussed and its implementation pipeline is presented which shapes the basic working structure of Visual Analytics of Simulated Data (ViDAS). Furthermore, the context of ViDAS is presented, keeping the human-centered design in mind. While the expert evaluation and feedback is based on simulation data, ViDAS is equivalently capable of handling usability data.

1.1 Motivation

With the technological advancements, data is automatically recorded and stored using various sensors and monitoring systems. This large amount of complex data is known as big data which is unstructured and contains hidden values. Therefore, there is a need to analyze this data, discover new values, and gain an in-depth understanding to efficiently manage and organize it [1].

Currently, there are a number of tools available in the market that help its users in analyzing the data and finding trends in it. These tools are mainly divided into two main categories; DM (Data Mining) tools and BI (Business Intelligence) tools. The DM tools are mainly focused on applying advanced machine learning models to the data and do not incorporate the advanced visualization techniques to go with it. The BI tools are mainly focused on EDA (Exploratory Data Analysis) techniques and include various clustering techniques along with interactive visualizations that help in understanding the data. However, the BI tools do not include built-in machine learning models and instead, rely on either third-party extensions or ask the user for a premium subscription.

This chapter presents a solution in the form of a web-based tool called ViDAS that incorporates the best features from both DM and BI tools. ViDAS combines these features in such a way that makes it effortless for the user to apply machine learning techniques on a complex data set, and then visualize the transformed data interactively.

1.2 Background

Intuitive computing technologies make their way into daily life and at the same time, the market is saturated with rival brands. This has made usability more popular in recent years, as businesses see the advantages of researching and designing their products using user-oriented approaches rather than traditional methods. Through knowing and studying the relationship between the product and the customer, the usability specialist may also have perspectives that are unfeasible by traditional market research. For example, after examining and evaluating customers, the usability specialist could recognize the requisite features or design shortcomings that were not expected.

Usability can be defined as the capacity of a software system to provide a condition for its users to perform tasks in a safe, effective, and efficient manner while enjoying user experience [2]. Usability requires techniques for assessing it, such as needs analysis [3] and the study of the values underlying perceived utility or beauty of the object. In the field of human-computer interaction and computer science, usability studies the sophistication and consistency with which interaction with a software system is built. Usability finds customer satisfaction and utility to be a quality component and strives to improve user experience through iterative design. Different researchers often focus on different parameters of usability, usability consultant Jakob Nielsen and computer science professor Ben Schneiderman have written (separately) about a framework of system acceptability, where usability is a part of “usefulness” and is composed of: Learnability, Efficiency, Memorability, Errors, and Satisfaction [4].

There are various methods that allow data collection of the above-mentioned parameters such as in-depth evaluations with a focus group, documenting the user experience, etc. Normally these parameters are analyzed using MS Excel and other traditional methods which require a lot of manual work. This is where the concept of visual analytics comes in as it provides an alternative approach that greatly supports the analysis process by the use of machine learning methods and detailed visualizations. Additionally, some of this usability data maybe complimented through simulations or machine learning methods that can be then analyzed to support the user-testing process.

In most cases, the dependencies and correlations of these parameters are not clearly identifiable, which forces the data analyst to make an educated guess. This guess is solely based on the expertise of the analyst which may result in extra time

and effort spent in testing the focused parameters. This approach is known as the “trial-and-error approach” [5] which focuses on finding a good solution and states that the data analyst must spend more time examining the parameters than building the model. In such a scenario, a better approach is to use visual analytics to understand the data better and find hidden relationships between the parameters. Visual analytics aims to help data analysts in identifying correlated parameters than relying on just the trial-and-error approach.

1.2.1 Visual analytics

Visual analytics is a human-centric process that combines techniques from graphics, visualization, interaction, data analysis, and data mining to support reasoning and sense-making for complex problems and extract relevant information from the raw data. While simple visualization techniques can be applied to the simulation data to investigate different parameters, find patterns, and visualize dependencies, data mining and machine learning helps with examining the data further through techniques such as forecasting, clustering, regression, etc. Visual Analytics is comprised of two parts; data analytics and data visualization. These two approaches co-exist and support the visual analytics process to understand complex data and find patterns in it.

Furthermore, visual analytics goes a step further and includes the approach of human-in-the-loop [6]. This approach combines the perceptual capabilities of a human mind along with the interactive data visualizations to apply visual analytics. Visual analytics is not a tool, but a human-centric process that aims at integrating human perception in to the visual data exploration process. It requires the specialist to first understand the data and its context. After the data is prepared, interactive visualizations can be used to find patterns and extract useful information from the data. An ideal visual analytics solution should require little to no coding, allow the user to combine from multiple sources, offer easy-to-customize interactive visualizations, include the feature to drill down the data at any level of detail, and combine multiple views to get an overall understanding of the data.

Currently, there is a lot of work being done in academia and industry towards visual analytics solutions to assist in the sense-making of the data [7]. There are a number of commercial business intelligence solutions that specialize in data discovery such as Tableau [8], Qlik Sense [9], Power BI [10], etc. Additionally, a number of data mining tools are available such as KNIME [11], RapidMiner [12], Orange [13], Weka [14], KEEL [15], etc. that focus on applying machine learning models and provide visualizations to help understand the raw data and find a pattern in it. However, there is a lack of tools that perform both data discovery and apply machine learning models.

1.2.2 Visual analytics pipeline

To apply visual analytics for both research and industrial applications, an appropriate definition and implementation of visual analytics pipeline must be followed that provides an effective abstraction for designing and implementing visual analytics systems [16]. The most common visual analytics pipeline can be seen in **Figure 1**. This conventional pipeline guides the visual analytics processes as an abstract outline which includes four major procedures and the relationships between them. This subsection explores these procedures in detail and discusses their role in the visual analytics pipeline.

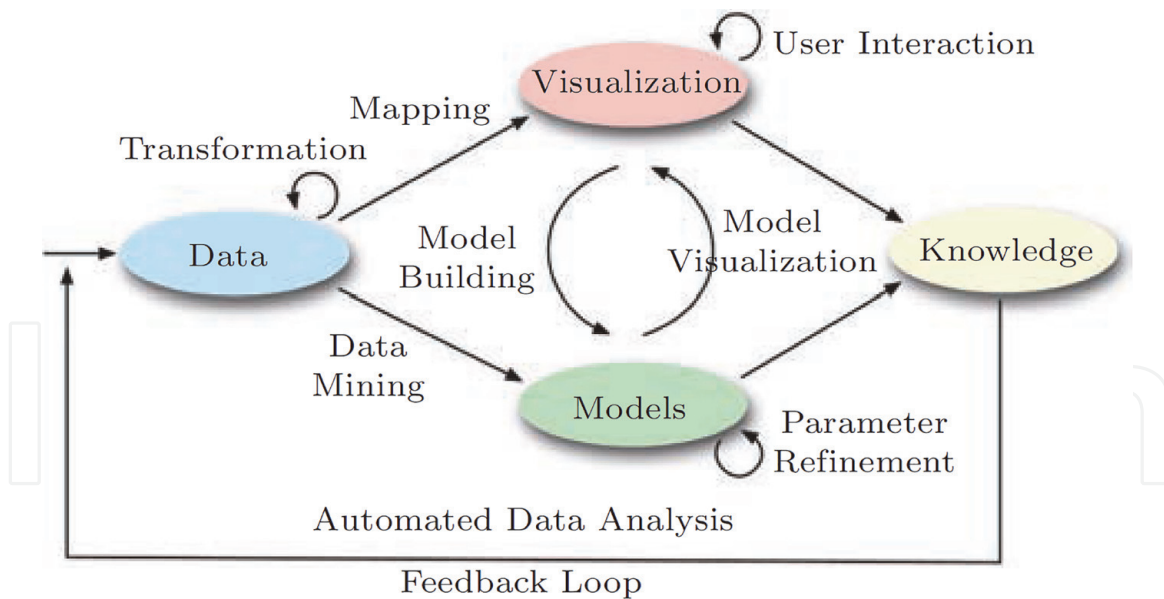


Figure 1.
Visual analytics pipeline by Keim et al. [17].

Data involves all the steps that are required to prepare the data set for visualizations and data analytics. These steps mainly include preparing and cleaning of data due to it being noisy and having missing values. All of these steps are collectively known as preprocessing steps and including data cleaning, data integration, data transformation, data reduction, and data discretization.

Visualization refers to the visual representation of data where the focus is on producing images that communicate the relationships between different attributes. This is achieved through the use of systematic mapping that establishes how the data values will be represented visually, determining how and to what extent a graphic mark property such as size or color will change to reflect the changes in the value of data. Effective visualizations help users to analyze the data and make complex data sets more accessible, understandable, and usable.

Model refers to the data analysis or machine learning methods that are applied to extract information out of the data which can later be visualized for crucial knowledge. These methods may include the steps from EDA in combination with machine learning algorithms such as Statistical models or Clustering models. After these methods have been applied, the resultant data is required to be visualized to extract knowledge that can help in understanding the data.

Knowledge refers to the process of generating a conclusion which either accepts or rejects a hypothesis. It is not a procedure by itself but an end result in the visual analytics pipeline. Knowledge is extracted from the data by applying all the above-mentioned procedures in order to understand the final visualization and get some insight into the data.

2. Related work

The relationship among all the procedures can be seen in the visual analytics pipeline i.e. visual mapping, data mining, model building, and model visualization. This section examines the different tools or techniques one would need to be able to perform the visual analytics pipeline tasks. As such Mapping and Model Visualization are grouped together under visualization tools while Transformation, Data Mining, and Model Building are grouped together under data mining tools. As

shown in the visual analytics pipeline, Visual Mapping is the relationship between Data and Visualization whereas Model Visualization is the relationship between Model and Visualization. In both cases, one would need to rely on data mining tools that apply machine learning models to the data and transform it. Additionally, visualization tools are required that focus on visualizing the data and model respectively.

2.1 Visualization tools

In contrast to open-source libraries for creating charts, there are also configurable (drag and drop) tools available for data visualization. In this context, we shall explore Tableau and Qlik Sense which are the current market leaders in data visualization and data presentation. These tools make it easier to extract and convey key patterns and insights, something that is important in both the Mapping and Model Visualization relations.

Tableau is one of the first tools we consider when we talk about commercial data visualization tools. It has connectivity to multiple data sources, many different switchable chart formats, and a sophisticated mapping capability that can easily convert simple Excel data to colorful dashboards with a lot of interactivity [8]. It has a step-by-step configurable interface from creating charts in sheets to filtering and combining multiple charts that form a Dashboard to overall storytelling.

Qlik Sense is a commercial data visualization and analytics tool that enables the user to import and aggregate data from diverse sources and use the data visualization tools of the software to convert raw data into meaningful information. It has an in-memory data storage engine which helps in dynamic visualization building [9]. Qlik Sense also follows a step by step procedure such as Tableau where each sheet (which is also a dashboard) may contain multiple charts; the sheets are then used to create a story by adding a snapshot of the charts or the complete sheet to the storyline.

2.2 Data mining tools

Another approach is to use Data Mining Tools in order to assist with the Data Transformation, Data Mining, and Model Building relations. There are various data mining tools available which are widely used by organizations for data transformation, data mining, and model building such as KNIME, RapidMiner, KEEL, Orange, WEKA, and many more. This subsection only discusses KNIME and RapidMiner as they are the most widely used data mining tools currently in the market.

KNIME [11] is an open-source data analytics tool that is written in Java and is based on the Eclipse platform. KNIME provides a user-friendly workspace and is based on the idea of graphical workflows to design the data analytics pipeline. It provides hundreds of nodes that incorporate data I/O, data cleaning, data manipulation, Machine Learning (ML) methods, scripting, and visualizations that can easily be used to create a workflow using a simple drag-and-drop approach [18].

RapidMiner [12] is an open-source tool that integrates a number of packages including text mining, ML, predictive analysis, DM, and business analytics [18]. Based on the tool, a desktop software is developed which is known as RapidMiner Studio which provides a GUI. With RapidMiner Studio, the user can perform DM and predictive tasks by creating workflows and then visualizing the output in an interactive representation [19]. RapidMiner allows scripting as well as workflows and is constantly being updated.

2.3 Summary state of the art

While all the tools mentioned above have their pros and cons, none of them completely cover the data analytics pipeline. The business intelligence tools are more inclined towards interactive visualizations and presentation of data, while the data mining tools are more focused on applying machine learning models to data and transforming it. ViDAS fills this void by combining the features of these tools and accommodating the complete visual data analytics pipeline.

3. User-context and requirements

Identification of user-context is the first step in a project based on human-centered design. It refers to understanding the users and identifying the intended way the project will be used by those users. It could be sufficient to just identify the stakeholders, but in most cases identifying the purpose and scope of the project help recognize the environment, it will be used in. Similarly in ViDAS development, the first step consisted of doing background research on the stakeholders, understanding their needs, and documenting the requirements. It was found that the stakeholders have a simulation model which generates data based on individual simulation runs. This data can not be easily understood by just looking at its tabular form, which is why advanced analytics is needed that extracts crucial information from the raw data and visualizes it in order to gather the knowledge. Once the context of ViDAS was understood, the next step was to gather the requirements for the development of the tool.

After identifying and understanding the context of ViDAS in terms of human-centered design, the next step is to gather and specify the requirements which will be the basis for the development and evaluation phases. The requirement gathering process is not as simple as discussing the needs of the stakeholders and documenting them. Instead, it includes making the stakeholders realize which requirements are needed according to the scope and context of the project [20]. The requirements of a project are further divided into three main categories; Business Requirements, User Requirements, and System Requirements [21]. In this section, we will focus on the business, user, and system requirements of ViDAS and discuss them in detail.

3.1 Business requirements

Business Requirements consist of the high-level requirements that answer generic questions to define the overall scope of the project [22]. These requirements also include the stakeholder's objectives and the needs of the target users for which the system is to be developed. The basic requirements were gathered from the initial communication with the stakeholders and stated that the users of ViDAS will be software engineers that will possess some basic knowledge of data analytics and visualization. The users will get the data from their clients and provide them with useful insights on raw, unprocessed data. For ViDAS, the business requirements specified that the user should be able to upload raw data into the tool, pre-process and transform the data, apply machine learning models, and visualize the data to extract useful knowledge. In addition to this, the user should also be able to get a general overview of the complete data set in the form of a visualization. The business requirements for ViDAS have been summarized in **Table 1**.

ViDAS Business Requirements
Apply data analytics on raw data
Visualize the data to get an overview
Apply machine learning models for data analytics
Pre-process and transform data according to the user's requirements
Visualize the transformed data

Table 1.
Summary of ViDAS business requirements.

3.2 User requirements

User Requirements are the specifications of how the user wants to complete certain tasks which are based on the business requirements of the project. User requirements include designing the layout of the system, the sitemap, and developing prototypes while keeping all the user goals in mind. With the help of user requirements, the user needs regarding how the system responds to user input are catered to. Once the business requirements were gathered, a meeting was arranged with the stakeholders which consisted of multiple steps including group discussions, workshops, and questionnaires to gather the user requirements and finalize the design of ViDAS. The main idea behind conducting workshops was to observe the stakeholders while using a similar tool and document their approach as well as the steps taken in a data analysis task. With the help of these workshops, important user requirements were identified and gathered.

There were two workshops conducted and the purpose of both workshops was to gather important feedback on how the stakeholders want to perform the data analysis and visualization steps. The first workshop focused on Tableau, which is a business intelligence tool that allows its users to perform data analytics and create interactive visualizations to find patterns in the data and extract useful information. The second workshop consisted of Qlik Sense and focused more on data analytics including applying machine learning models on the data and later create interactive visualizations. The task lists designed for both of the workshops were fairly similar yet focused on their respective purpose and objectives. Crucial information was gathered during these workshops and important user requirements were pointed out during the discussions. After the workshops, the stakeholders were asked to fill questionnaires that are summarized in **Table 2**, giving brief feedback regarding the required user interaction and user experience the stakeholders expect.

With the help of the workshops and the questionnaires, the user requirements of ViDAS were collected. The user requirements focused on the drag-and-drop implementation of the tool during the data analytics and visualization processes. The custom analysis features of Qlik Sense and Tableau did not cover the basic requirements, due to which the custom analysis was prioritized to give ViDAS an advantage over these tools. The custom analysis should be created using a workflow pipeline and the resultant data should be exported to the visualization tab. The workflows in the custom analysis should comprise templates containing advanced machine learning models, and the ability to save and load custom workflows. Furthermore, the user should be able to write custom python scripts and execute them. The visualizations should be interactive and created using the charts as well as the fields. In the case of fields, ViDAS will automatically detect the dropped fields and create the best-suited chart based on those fields. The user requirements for ViDAS have been summarized below in **Table 3**.















Evaluation Summary		
	Qlik Sense	Tableau
Visualizations		
Data Overview		
Chart-building Process		
Chart Recommendations		
Interactive Visualizations		
Custom Analysis		
Machine learning modules		
Visualize transformed data		
Overall		
Overall Tool Execution		

Table 2.

An overview of Qlik sense and tableau evaluation (1 star - unforgettably bad, 2 stars - below average, 3 stars - average, 4 stars - above average, and 5 stars - unforgettably good).

ViDAS User Requirements
Interactive visualizations
Chart Recommendations
Drag-and-Drop design of the tool
Workflow pipeline for data analytics which is created using the drag-and-drop approach
Save & Load workflows
Pre-designed workflow templates
Custom python scripts for data analytics

Table 3.

Summary of ViDAS user requirements.

3.3 System requirements

System Requirements are the low-level requirements that act as the basic building blocks on which the system will be developed. These requirements cover all the technical details of the project including the technology used, the versioning, the compatibility, the database, and if the system will be hosted on a server. Once the user requirements were finalized, the next step was to define the system requirements based on the gathered user requirements. The discussions after the workshops gave a broad idea regarding the technology to be used in ViDAS development, thus shaping the system requirements as seen in **Figure 2**. It was required for the tool to be integrated into an already built website which was developed using React.js as a front-end and Java Spring as the back-end. Due to this limitation, ViDAS was

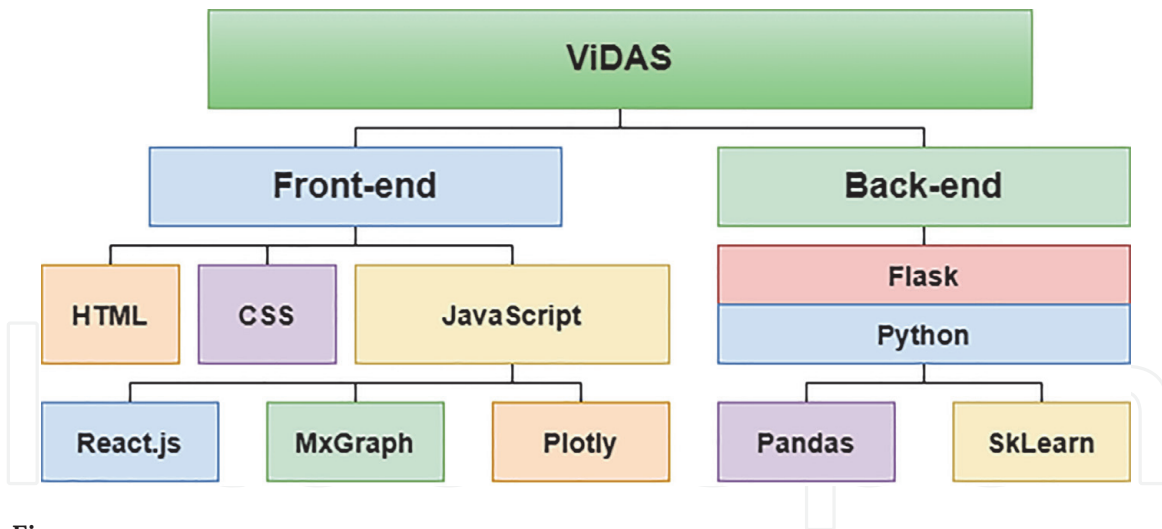


Figure 2.
ViDAS technology stack.

required to be developed using React.js as the front-end, while there was no restriction regarding the back-end implementation. After a detailed research, Python was decided as the back-end language due to it being highly dynamic for performing data analytics on big data and packaging a number of machine learning libraries. Flask was decided as the web framework for python to connect and communicate with the front-end.

For the front-end module, basic HTML was used to define the structure of ViDAS and CSS was used to style the HTML components for a complete user experience. React.js was used as the JavaScript framework to define the support of the functionalities in the front-end. To cater to the drag-and-drop requirement, MxGraph was used and customized according to the user requirements of ViDAS. The use of customized MxGraph further supported the User Interface and increased the intuitiveness of the tool while boosting the User Experience. Plotly was used to create interactive visualizations. These interactive visualizations were created in the back-end module and transmitted to the front-end for displaying them to the user.

The front-end was developed to run on the user's browser whereas the back-end was implemented to run on the server. The communication between the client and server was done using a promise-based HTTP API called Axios. With the help of Axios, the communication data was transmitted between the front-end and the back-end in JSON format. A state management library called Redux was used to manage the state and store the important data that was used by multiple pages on the front-end. Due to Redux, the data transmission among the front-end pages was eliminated and stored in a centralized state container.

The back-end is comprised of a number of Python libraries to deal with the big data and the data analytics functionalities. Pandas was used to store the big data in tabular form which helped in performing pre-processing steps before applying the machine learning models. Scikit-learn, a machine learning library, was used to support the machine learning models and provide various data analysis methods. The system requirements for ViDAS have been summarized below in **Table 4**.

3.4 Requirement outcomes

This subsection gives a brief outcome of this section and discusses the business, user, and system requirements that were gathered. The business requirements were gathered during the initial communication with the stakeholders and as a result, the scope of ViDAS and its users were identified.

ViDAS System Requirements
React.js as front-end language
mxGraph for drag-and-drop implementation
Redux for state handling
Python as back-end language
Flask as python web framework
Pandas to store big data
Scikit-learn for machine learning models
Plotly for data visualizations

Table 4.
Summary of ViDAS system requirements.

To gather the user requirements, two workshops were designed that comprised of task sheets. The stakeholders were guided through the tasks and their actions were observed. A number of new requirements were identified during this process. After the workshops, the stakeholders were asked to fill the open-ended and closed-ended questionnaires that gave shape to the user requirements of ViDAS. Once the user requirements were finalized, a number of post-workshop discussions were arranged where the technology stack of ViDAS development was discussed which gave shape to the system requirements of ViDAS.

All the requirements were gathered over a series of meetings, workshops, and discussions which have been summarized above in **Tables 1, 3, and 4**. Now that the requirements of ViDAS were finalized, the next step in the human-centered design process was to produce the design workflows and to implement.

4. Producing design solution

The design creation process can be conducted in different ways depending on the scenario, from copying and development from previous design inspirations, to creating innovatively. Regardless of the source, all the design ideas progress through iterative development in the human-centered design approach. In such cases, mock-ups and simulations are essential to support this iterative cycle. Various design techniques are available such as brainstorming, parallel design, storyboarding, paper-based prototyping, and computer-based prototyping. While it is not intended to imply that all these techniques should be used in every product development, they should at least consist of a series of UI (User Interface) screens, and a partial database which allows the user to interact, visualize, and comment on the future design [23]. These early simulations are easy to create and result in a fault-free product at the end. The experts, stakeholders, and user representatives in the design development cycle help identify the faults, correct design, and undergo the costly process of re-implementation once the design is finalized.

After the requirements were finalized, as discussed in section 3, the next step was to create the design of ViDAS. Paper prototyping was used as a starting point to create the conceptual design of the tool. The stakeholders were involved in the design process to get their important feedback while creating the design. After the paper prototype feedback, the Balsamiq wireframing tool was used to transfer the conceptual design into interactive mock-ups [24] to better communicate the design with the stakeholders. These mockups consisted of design for each page of the tool

and interactivity within the page (buttons, hyperlinks). Once the design was created and approved by the stakeholders, the development of ViDAS was started.

This section is divided into two subsections. The first one discusses the paper prototype of ViDAS. The next subsection addresses the software prototype and the implementation of ViDAS. These subsections will also address different tools that were used during this iterative process.

4.1 Paper prototyping

Paper prototyping is a widely used approach in the human-centered design process. It is a throwaway prototype technique used to create the initial conceptual design for a tool or application. Paper prototypes involve creating rough, even hand-sketched drawings or models of a design. The functionality is simulated by a member of the design team playing the computer and responding to the user's inputs by swapping the bits of the paper or writing an output. Creating paper prototyping is simple; however, it can provide convenient stakeholder feedback to aid the design [25].

Based on the requirements, paper prototypes of the different elements were drawn using pen and paper. These paper prototypes consist of data processing, chart creation workflow, and analytics workflow. Also, all the other necessary elements, such as menus, icons, buttons, labels, and dialog sequences were drawn. **Figure 3** shows the ViDAS paper prototype; each view has a description of what can be done and what happens when one interacts with individual elements.

The testing of the paper interface was video-taped as the elements moved and changed. This videotape of the paper prototype was shared with the stakeholders via email to get feedback about the initial design. Paper Prototyping was a handy technique during ViDAS initial design creation and getting the stakeholder's thoughts about the tool's overall shape. However, during the post-evaluation discussion of the paper prototype, it was found that the paper prototyping approach

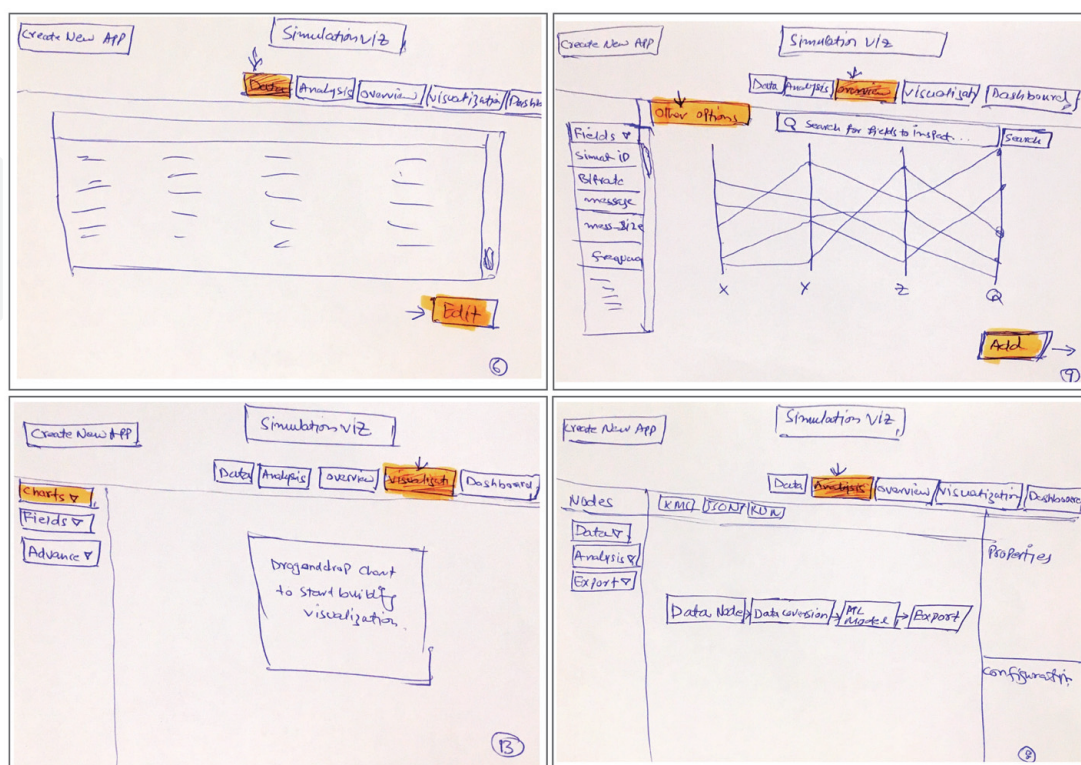


Figure 3.
ViDAS paper prototype.

does not discover every single usability problem. The paper prototype of the initial design was transformed into interactive mockups to address every aspect and give the design a more realistic feeling. It was also beneficial to communicate the design better with the stakeholders.

The mock-ups of ViDAS were created using Balsamiq [24] wireframing tool which is an industry-standard, light-weight wireframing tool used to create design mock ups and to show the interactivity among different pages and elements of the design. These mock-ups consist of each page of the tool along with interactivity that redirect when buttons or hyperlinks are clicked.

Figure 4 shows different design mock-ups of ViDAS. The Data tab view shows the tabular view of the uploaded file. Next to Data, the Overview tab is about visually inspecting the overall data. In contrast, the the Data Analysis tab shows the chart creation process that focuses on data fields. The Custom Analysis tab shows the workflow of creating analytics using network graphs. In this phase of the design, different changes were carried out which are summarized in **Table 5** and discussed later on.

ViDAS Mock Ups Evaluation.

Different changes were made to the design in this stage of the design creation process. The mockups creation process was iterative, and the focus group members were involved throughout this process. Mockups for different pages and interactivity among them were created, and the suggested changes were added iteratively until the final design was approved.

Table 5 shows all the stakeholders' suggested changes. These changes were added or deleted in multiple iterations in the ViDAS design. The label "Included" or "+" refers to components added to the design, while "Excluded" or "-" refers to the design components excluded from the final design. Some of these changes included

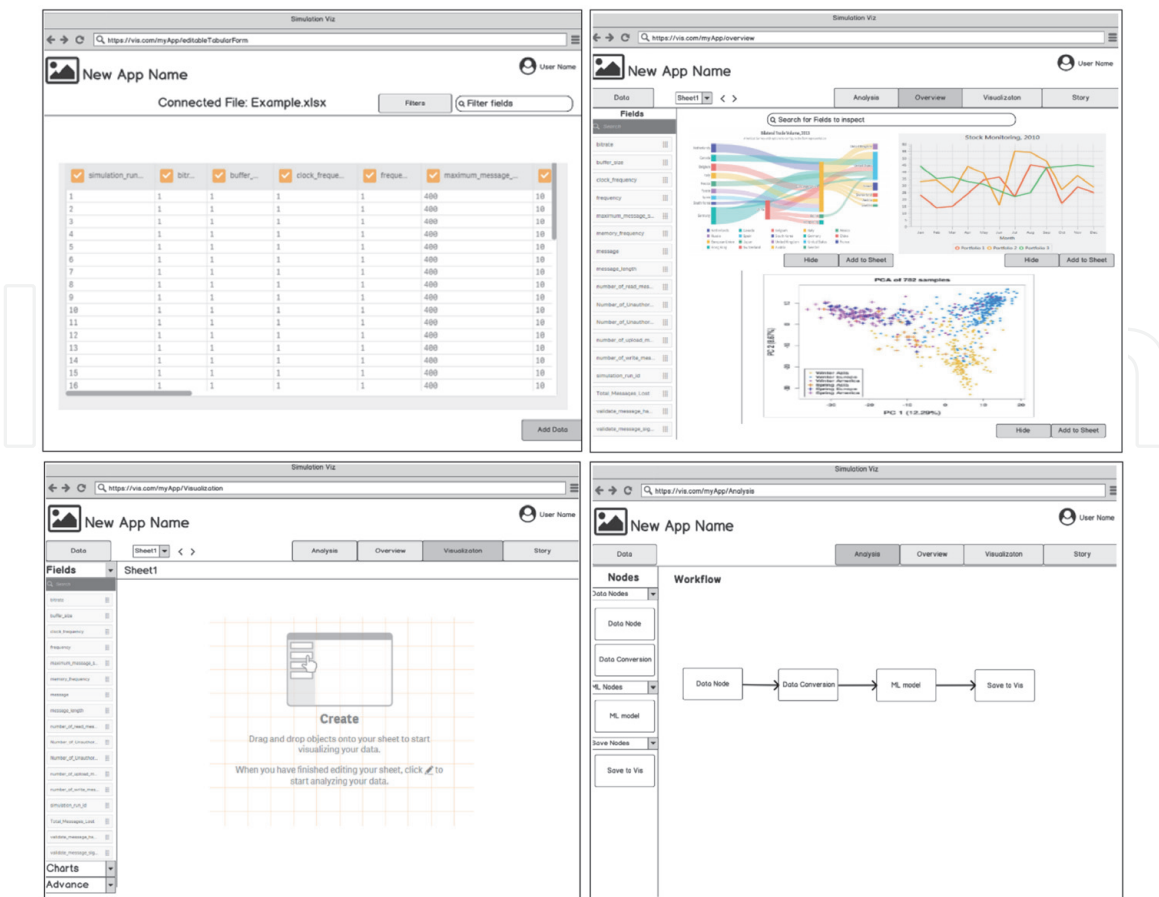


Figure 4.
ViDAS mock ups.

ViDAS Design Changes (– = Excluded), (+ = Included)			
Design Components	First Iteration	Second Iteration	Final Design
Login System	+	—	—
Color Combination	+	+	+
Button Labels	+	+	+
Tabs Concept	—	+	+
Sheet/New Sheet	—	+	+
Multiple Data Source Upload	+	—	—
Labels	+	+	+
Data Filtering	—	+	+
Data Tab	+	+	+
Overview Tab	+	+	+
Data Analysis Tab	+	+	+
Custom Analysis Tab	+	+	+
Story Tab	+	—	—
Dashboard Tab	—	+	+
Chart Per Sheet	—	+	+
Chart Creation (Drag and Drop)	—	+	+
Create Chart (Click and draw)	+	—	—
Create Chart (Selected Fields)	+	—	—
Drop-down menus	+	+	+
One Chart Per Sheet	—	+	+

Table 5.
Iterative ViDAS mock ups.

the User login system that was a part of the design but later excluded because ViDAS was planned to be an integrated module to a previous application and that application already has a user login system. Upload functionality was changed because currently the stakeholders use Excel data format for simulation output. Another design component was to create visualizations from selected fields exported from the Overview tab but was also excluded because the Overview Tab is currently supposed to only find trends and patterns in the uploaded data. Inspired by the tableau design, the concept of one chart per sheet (which would later be used for dashboard building) was also included in the design. Design for chart recommendation and sheets was also included later on. Further, changes such as labels of tabs, title page, and buttons were changed throughout this phase of the design.

4.2 Software prototyping

Once the requirements and design for ViDAS were finalized, the next step was to develop the tool or software prototype. ViDAS implementation covered all the requirements that were gathered in the requirement gathering process. The technologies that were used in ViDAS development allowed it to be integrated into the existing systems while still being easily extensible. ViDAS is implemented using the concept of RESTful APIs, where the front-end and back-end modules are separate

and communicate with each other by sending the data over the network using their respective web addresses.

This section addresses the implementation of different components of ViDAS. It also discusses how and which component of the tool covered the requirements, such as applying analytics on raw data, visual overview of the uploaded data set, data preprocessing, and creating interactive visualizations.

The **front-end** of ViDAS is designed in such a way that when a user switches between different tabs, it keeps the previous page state intact. **Figure 5** shows different tabs of ViDAS. The **Data** tab is the entry point to ViDAS. When the file is uploaded, the tool shows the data in a tabular view initially. Thereafter the **Overview** tab gives a visual overview of the uploaded data. The **Data Analysis** tab handles the chart creation and chart recommendation process, and the **Custom Analysis** tab covers the data analytics process.

4.2.1 Data tab

Once a file is uploaded, the user is shown the uploaded file's tabular view. Data filtering can be carried out to delete unnecessary columns by clicking the **Edit** button at the bottom of the tab. If the data is clean and does not need filtering, the user can either switch to the Overview tab to see the data visually and find trends and patterns in the data, directly create charts by clicking the Data Analysis tab, or perform analytics on the data in Custom Analysis tab.

4.2.2 Overview tab

The overview tab covers the requirement "Visualize data to get an overview". This tab comprises three high-level charts of the uploaded data, as shown in **Figure 6**. The main objective of the Overview tab is to find trends and patterns in the data using a single interactive view.

The design of the Overview tab is divided into two sections. The left **side bar** consists of a drop-down for the high-level charts (Parallel-coordinates, Sankey

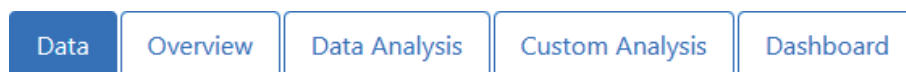


Figure 5.
ViDAS tabs.

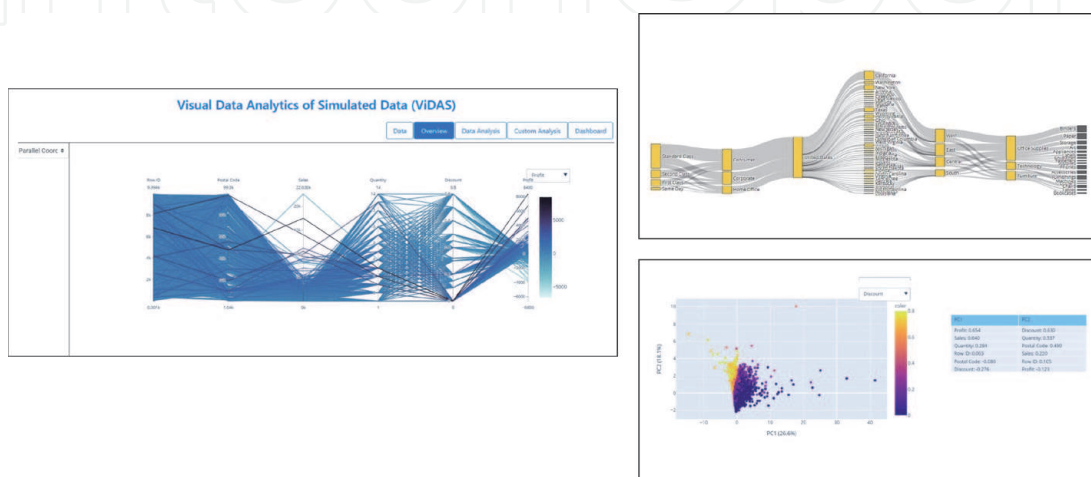


Figure 6.
ViDAS overview tab.

diagram, and PCA (Principal Component Analysis)), whereas the main canvas in the middle of the tab shows the selected type option from the drop-down. When the Overview tab is clicked, Parallel-coordinates is shown by default. Optionally, the user can switch to another chart option from the drop-down list. Here, Parallel-coordinates allows comparison of the individual observations (series) on a set of numeric variables. While Sankey diagram is a visualization technique that allows displaying data flow and PCA indicates variation and brings out strong patterns in a data set. It is often used to make the data easy to explore and visualize.

4.2.3 Data analysis tab

The Data Analysis tab addresses the requirements related to “Creating interactive visualizations” as stated by the stakeholders including chart creation by drag-and-drop, chart recommendations, and creating a chart from the exported custom fields from custom analysis tab.

The front-end of the Data Analysis tab is divided into different sections, as shown in **Figure 7** which consists of a sidebar, a toolbar, sheets on the bottom, and the recommendation charts drop down on the top right side. The **left side bar** comprises of all the drag-and-drop objects, such as standard and advanced charts, uploaded file’s fields, and the custom fields exported from Custom Analysis tab. Similarly, the **top toolbar** consists of a text field which is used to rename or delete the active sheet and the **top right drop-down menu** is for chart recommendation functionality. The **Wrapper canvas** in the middle of the view allows the chart/fields to be dropped for chart creation. Furthermore, the **footer** holds all the sheets created by the user. The newly created data attributes which are exported from “Custom Analysis Tab” are shown in “Custom Fields” and can be used together with other fields for chart creation.

Charts in ViDAS are categorized into standard charts (e.g. line chart, bar chart, box plot, bubble chart, etc.), advanced charts (linear regression and correlation matrix), and overview tab charts (parallel coordinates, sankey chart, and PCA). Chart creation in the data analysis tab can be done in one of two ways. It starts when the user drops an object (chart/field(s)) to the canvas and provides the required parameters. The tool then generates the required chart for the user based on the provided data.

Create Chart By Chart Drop: Once a certain chart is dropped to the canvas, ViDAS first validates the type of the dropped object (chart/field(s)) and then

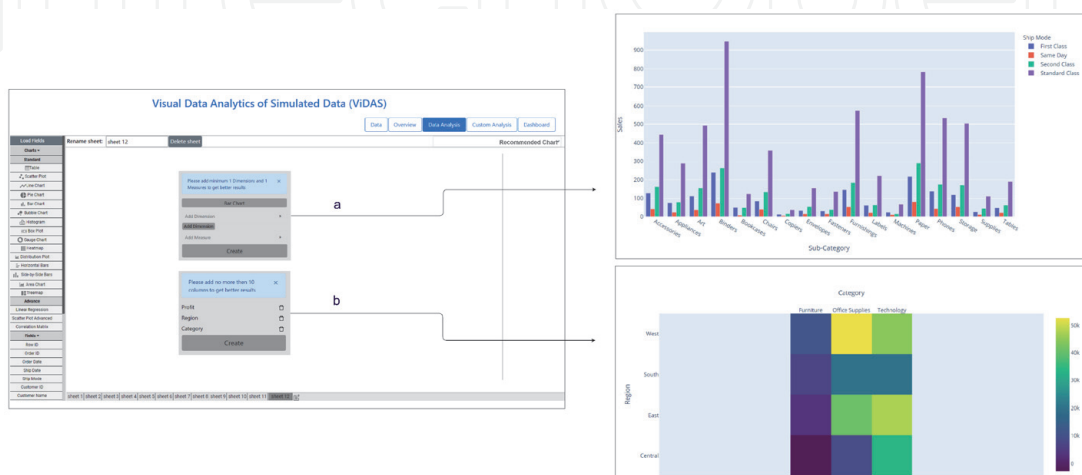


Figure 7. ViDAS data analysis tab: (a) represents chart creation using chart drag-and-drop. (b) Represents chart creation using fields drag-and-drop.

renders a cart. The cart shows the dropped chart name on the top and a validation message about the required minimum parameters for the specific chart type for sensible chart creation. Once the required parameters are added and **create** button is clicked, the required chart is created.

Create Chart By Field(s) Drop: The chart creation by field(s) drop is similar to the chart drop. However, in the field(s) drop scenario, the user only drops the interesting field(s) as shown in **Figure 7**. ViDAS automatically calculates the type of the dropped field(s) as dimensions or measures. Thereafter, similar to the chart drop validation the user is shown a validation message with all the dropped field(s) for creating a proper chart. Here, the user may delete from the dropped field(s) as well. Once all the desired field(s) are dropped and **create** button is clicked, the tool creates the best possible chart by using the “Chart Recommendation” feature. At this point, the user can add additional fields or delete from the selected fields to update the created chart. The chart will change instantly and allows the user to see the changes visually while adding or removing certain fields.

Chart Recommendation: Once a chart is created, the user can switch to other possible charts in the “Recommended Charts” drop-down menu. This feature is known as ‘Smart Charts’ which was implemented in python and the resultant charts were created using the Plotly charting library. The feature collectively dubbed ‘Smart Charts’ is a collection of few chart recommendation methods. Once a chart is created, it suggests other possible charts that could potentially be made with the same fields that the user used to create the previous visualization so as to explore alternative charts that could better represent the data at hand.

The basic working of ‘Smart Charts’ depends on the attributes selected and their data types. Dimensions are the qualitative aspect of the data, in other words, the categorization and context of data whereas measures are the actual values of what you are measuring. Measures are always numeric and they are, most of the times, aggregated in some way using an aggregation function like sum, average, etc. The chart recommendation drop-down list contains all the charts for which the requirement of minimum dimensions and measures is fulfilled. The order in which they are displayed is decided by the weights that have been manually assigned to them. These weights are decided on basis of analysis of rival industry-standard data analysis tools, trial-and-error, and testing. A general pattern is, the more ‘constrained’ requirements that a chart has, the higher the weight it gets assigned. In other words, the more niche a chart is, the more important it would be if it can be created at all.

These recommendations in the “Recommended Charts” drop-down menu are listed in descending order of priority i.e. the higher up the entry, the better it is assumed to be. But that might not always be true, so it is up to the user to decide if the recommended charts present their data to their liking or not. ‘Smart Charts’ is a single method which comprises of various sub-modules including ‘All possible charts’, ‘Best possible chart’ and a ‘Drag-and-Drop Chart Generator’.

If the user has an idea of what they want to create, they can directly choose a chart type, give it the required inputs and the specific chart would be generated. But if the user does not know what to create and they are roughly aware of the data fields in question, they can drag and drop the specific fields instead of the chart type. From there, ViDAS infers the dimensions and the measures to render the ‘best possible chart’.

4.2.4 Custom analysis tab

The Custom Analysis Tab addresses the requirements related to the data analytics part of ViDAS that includes creating machine learning models for data analytics,

pre-processing, and transformation of the data according to the user's requirements, and exporting the transformed data to the "Data Analysis Tab" for visualization.

Figure 8 shows the "Custom Analysis Tab", which consists of a **Side Bar** that contains all the drag and drop nodes. The user drags and drops these nodes onto the Canvas, which then creates a graphical representation of these nodes and can be connected later from their ports using edges. The **Properties Bar** shows the details of the selected Node, including the ID, Node Type, Node Name, the input, and output types. Once a node is configured, the user-defined configuration in the **Configuration Bar** is displayed. The **Tool Bar** includes the usability functionalities to help the user, including the option to save and load the workflows to and from the server, deleting the workflows, and creating a new workflow to clear the canvas.

The basic node design can be seen in **Figure 9**, with one input port and one output port. On top of the Node, there are details such as the node type and the custom name. The node type **Custom Node** is dynamic and changes according to the Node used. The node name **customName** can be changed by double-clicking on the Node. A square represents the input ports, and a triangle represents the output ports. At the bottom of the Node, there is an overlay that shows the current node state by using color indicators. When the user right-clicks on the Node, several options are available in the form of a drop-down menu. These options include Configure, Run, View Output, and Delete Node.

Initially, all the nodes are unconfigured when dropped, which can later be configured before being executed. ViDAS will not allow a node to be executed if that Node or any of its parent nodes are unconfigured. Some nodes require access to the data from their parent nodes to be configured, due to which they cannot be

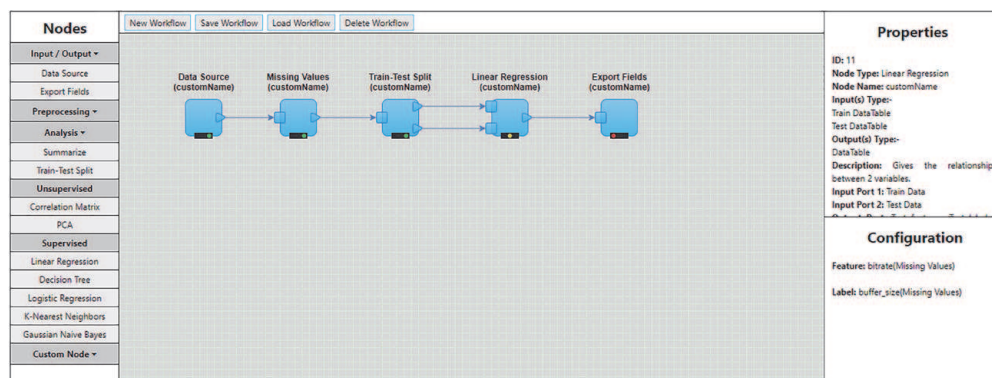


Figure 8.
ViDAS custom analysis tab.

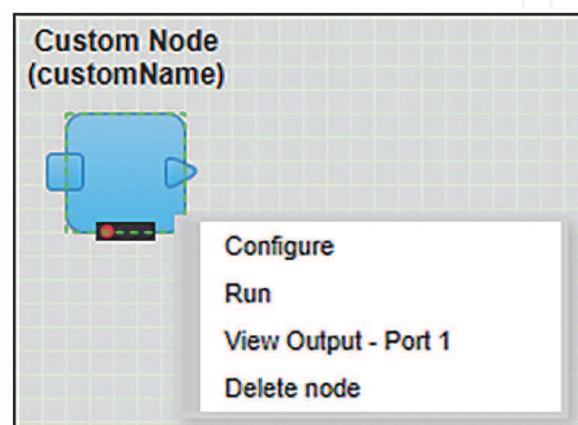


Figure 9.
Basic node.

configured unless the parent nodes are executed first. If multiple nodes have been configured in a workflow, the user can execute only the last Node, which will recursively execute all of the previous nodes up to and including that node.

ViDAS also allows the user to write custom python scripts to apply data analysis techniques that are not found in the built-in nodes. The user can select from a wide variety of node categories including input/output, preprocessing, analysis, and custom node to create the workflow pipeline.

Once the workflow pipeline is complete, the user has the option to export the transformed data to the “Data Analysis Tab” using the Export Node. The Export Node can be configured to select the required columns that the user wants to visualize. These exported columns can be loaded in the ‘Custom Fields’ drop-down list during data visualization. The feature to export the transformed data to another Tab gives ViDAS an integrated user experience which comprises of both data mining and interactive visualization methods.

5. Evaluating design solution

Evaluation is one of the most important components of project development and can play an important part in keeping the project aligned with its scope and objectives. Evaluation is used to support the decision making in development by applying evaluation methods and gathering the important data to compare the project to its pre-defined goals.

There are a number of evaluation types that exist but they can all be summarized into three most basic types which are Goal based, Process based, and Outcome based evaluations.

Goal based evaluations are mainly focused on the objectives of the project that have been pre-defined during the requirements gathering process. Once the project development is complete, it is evaluated to see if the features included in the project support the goals and objectives that were pre-defined.

Process based evaluations focus on the project’s quality, strengths, and weaknesses. It discusses if the processes included in the project satisfy the stakeholders and are implemented the way they were intended. It also summarizes the strengths and weaknesses of the project as important feedback to improve in the next iterations.

Outcome based evaluations discuss the lasting effects of the project and the greater good that can be served as an outcome of the project. It measures the final goal of the project and how well it has been achieved.

This section will discuss ViDAS evaluation. It will also discuss the steps that were taken to evaluate ViDAS and to obtain important feedback, which was then used to improve the tool.

5.1 ViDAS evaluation

Once the implementation was done, the next step was to evaluate the ViDAS development. This evaluation was based on the business, user, and system requirements that were gathered during the requirements elicitation process discussed in section 3. First, the tool was self-evaluated and later by our stakeholders and users to get their useful feedback to improve the User Experience and User Interface of ViDAS.

The evaluation of ViDAS was mainly goal-based as the development was done keeping in mind the requirements and objectives that were defined before the implementation. Various evaluation methods were combined for ViDAS as shown

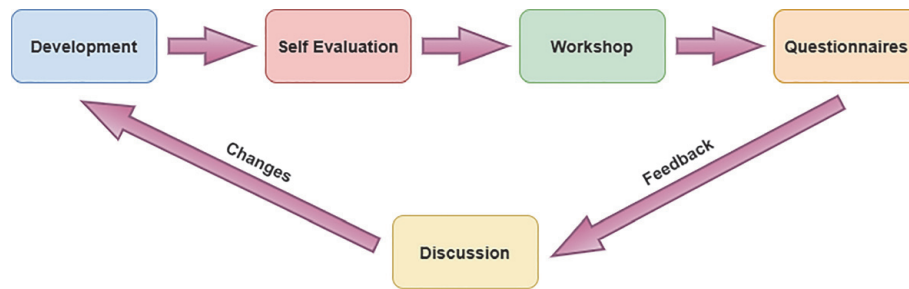


Figure 10.
ViDAS evaluation steps.

in **Figure 10**, including comparing the goals to the pre-defined documents, arranging a workshop with the focus group, and collecting important data before and after the workshop. The evaluation also partially included segments from process based and outcome based evaluations as the project's quality was assessed, important feedback was collected, and the final goal was studied during the complete process.

Once the development of ViDAS was completed, the features and functions developed were compared to the pre-defined requirements. The product was assessed in the view of the objectives and goals defined during the requirements gathering process. Also, a number of tests were carried out that included analyzing the reliability of the front-end on multiple browsing agents as well as various screen sizes. Stress tests were also a vital part of the self-evaluation process and ViDAS was tested by uploading big data containing millions of rows to assess the data handling time and look for any bottlenecks that may hinder the tool's performance. All of these steps were an important part of self-evaluation and the necessary changes were made to improve the final User Experience (UX) before preparing a workshop with the stakeholders.

After self-evaluation, a workshop was arranged with the stakeholders. This workshop was prepared using tasks similar to the ones used during the requirements gathering process to compare and improve with the state-of-the-art in the market. Similar tasks made it easy for the stakeholders to compare ViDAS with the tools used in the requirements gathering workshop.

5.2 Feedback

After the evaluation workshop, the initial feedback was collected from the stakeholders and documented to improve ViDAS. The stakeholders were pleased with the overall implementation of the tool. While there were suggestions to improve ViDAS, it was recognized to be a great help with the analysis of the simulation data for the stakeholders.

The post-workshop feedback was gathered by asking the stakeholders to fill open-ended and closed-ended questionnaires that have been summarized in **Table 6**. Overall, the stakeholders were satisfied with the tool execution. The data overview, the chart building process, and the interactive visualizations were greatly appreciated. The stakeholders also found the chart recommendation concept useful. The custom analysis features were commended and the workflow pipeline design which comprised of the drag and drop module was praised by the stakeholders. Furthermore, the development of ViDAS in such a short span of a few months was greatly appreciated, which resulted in the overall feedback being much positive. However, the overall tool execution feedback was lower than anticipated and this was mainly due to some usability aspects such as missing validations, missing tool tips, and labels not being user-friendly. These usability issues have since been resolved and have been implemented in the second iteration of the tool.

ViDAS Evaluation Summary	
Visualizations	
Data Overview	★★★★☆
Chart-building Process	★★★★★
Chart Recommendations	★★★★☆
Interactive Visualizations	★★★★☆
Custom Analysis	
Workflow Pipelines	★★★★★
ML modules without coding	★★★★★
Export Data for Visualizations	★★★★★
Overall	
Overall Tool Execution	★★★★☆

Table 6.

ViDAS evaluation feedback overview (1 star - unforgettably bad, 2 stars - below average, 3 stars - average, 4 stars - above average, and 5 stars - unforgettably good).

6. Conclusion & future work

6.1 Conclusion

The demand for a good visual analytics tool is increasing due to the need to cater to complex data which is constantly increasing. A complete visual analytics solution should cover all the procedures in the conventional visual analytics pipeline. These procedures, starting from the basic data pre-processing to clean the data, allow the user to apply machine learning methods in order to transform the data, and extract hidden correlations in the data. Later, the transformed data can be visualized to find trends and patterns that can be crucial in extracting knowledge from the data.

While there are various tools available in the market that allow the users to either create dynamic, interactive visualizations or implement machine learning models, none of the tools cover the complete visual analytics pipeline with their built-in features. These tools rely on third-party extensions that can be quite tiresome to configure and integrate. Additionally, integrating these third-party extensions may make the tools quite heavy and unstable. This project discusses the visual analytics pipeline and presents a solution that implements the complete visual analytics pipeline in order to fill this void. In this project, a web-based tool called ViDAS was developed using the “Human-centered approach” that packages the complete visual analytics pipeline integrated into it. ViDAS was developed with a number of requirements in mind that are not available in the tools currently in the market. These requirements included the tool being light-weight so that it can be deployed on a web server, dealing with big data, allowing machine learning methods and custom scripts, and visualizing the data in an interactive way.

Initially, the context of use for the tool and requirements were gathered in a series of workshops and meetings. ViDAS allows its users to upload raw data and apply initial data filtering to only process the required attributes. These attributes can later be used to either create interactive visualizations or apply custom analysis.

If the custom analysis is applied, the resultant attributes can be exported to the 'Data Analysis' tab for visualization of those ML models. The chart creation process of ViDAS comprises of a drag-and-drop workflow and is very intuitive. Furthermore, ViDAS packages a chart recommendation functionality that suggests a number of charts according to the fields selected. By comparing the feedback of the tools used in the requirement gathering process to the feedback of ViDAS in its evaluation, an improvement was seen in the overall tool experience.

6.2 Future work

In this project, a complete visual analytics tool was presented that can help the users in finding hidden relationships in their data by providing various machine learning and visualization techniques. However, the tool is still in its initial phase and requires more features in order to cater to all the needs of a data analyst.

In the initial ViDAS implementation, there was an intention to include dashboards, which would feature the coordinated views, cross-filtering i.e. filtering/selecting data in one chart would reflect relevant changes in other linked charts. Dash, an extension to the Plotly charting library, was explored to create these coordinated views but unfortunately, this could not be completed due to time constraints and may be added in the future.

Another interesting feature that could be implemented in the future is the functionality to select desired fields in the overview tab. During the overview tab view, the user may find multiple correlated fields that are important to visualize. These fields can be intuitively selected from the overview visualization to be used in the desired visualizations. Furthermore, an area that can be significantly improved further is the processing of date/time data during the visualizations. Currently, dates are treated as any other dimension or ignored. There could possibly be sophisticated parsing by using parts of dates such as years and months. Also, interactivity within the visualizations can be further improved as Plotly features a handful of interaction widgets that can be added.

The functionality of 'Smart Analytics' can be integrated into the tool to help the user in applying the machine learning models. In some cases, the raw data can be unstructured and complex which may pose a challenge in deciding which machine learning models to apply. In such cases, the smart analytics feature can use data mining techniques to explore the type of data in the data set and suggest the recommended machine learning models to apply based on the data type and complexity. Additionally, more ML nodes can be introduced that offer flexibility in dealing with complex data. Forecasting can be used with time-series data to predict future values depending on the past data-set.

Furthermore, due to the current situation of a pandemic, the evaluation of ViDAS could not be performed to the full extent. The evaluation was conducted by using only the stakeholders as the focus group. In the future, a complete evaluation should be conducted that includes a mixed focus group comprising of the domain experts as well as the non-technical users.

These new features combined with the in-depth evaluation of ViDAS can result in a better and thorough visual analytics tool that provides a user-friendly experience and helps its users in dealing with raw and complex data by allowing data analytics as well as interactive visualizations.

Acknowledgements

The authors wish to thank the members of the Human Computer Interaction Lab at the University of Kaiserslautern and the Embedded Software Engineers at the

Fraunhofer IESE for their cooperation. This work was supported by the Ministry for Science, Education, and Culture Rhineland-Palatinate through the Virtual Engineering of Smart Embedded Systems (ViSE) project.

IntechOpen

IntechOpen

Author details

Taimur Khan^{*†}, Syed Samad Shakeel[†], Afzal Gul[†], Hamza Masud
and Achim Ebert[†]
University of Kaiserslautern, Kaiserslautern, Germany

*Address all correspondence to: khan@rhrk.uni-kl.de

† These authors contributed equally.

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Chen M, Mao S, and Liu Y. Big data: A survey. *Mobile networks and applications* 2014; 19:171–209
- [2] Lee JY, Kim JY, You SJ, Kim YS, Koo HY, Kim JH, Kim S, Park JH, Han JS, Kil S, et al. Development and Usability of a Life-Logging Behavior Monitoring Application for Obese Patients. *Journal of Obesity & Metabolic Syndrome* 2019; 28:194
- [3] Smith KT. Needs analysis: Or, how do you capture, represent, and validate user requirements in a formal manner/notation before design. *Human Factors and Ergonomics in Consumer Product Design: Methods and Techniques* 2011; 415
- [4] Nielsen J. Usability 101: Introduction to usability. 2012. Available from: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [5] Kleijnen JPC, Sanchez SM, Lucas TW, and Cioppa TM. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS J. on Computing* 2005 Jul; 17:263–89. DOI: 10.1287/ijoc.1050.0136. Available from: <https://doi.org/10.1287/ijoc.1050.0136>
- [6] Kehrer J and Hauser H. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 2013 Mar; 19:495–513. DOI: 10.1109/TVCG.2012.110
- [7] Post T, Ilsen R, Hamann B, and Aurich J. User-Guided Visual Analysis of Cyber-Physical Production Systems. *Journal of Computing and Information Science in Engineering* 2016 Oct; 17. DOI: 10.1115/1.4034872
- [8] Tableau. Business Intelligence and Analytics Software. <https://www.tableau.com/>. [Online; accessed 25-February-2020]
- [9] Intelligence QB. Data Analytics and Data Integration. <https://www.qlik.com/us/>. [Online; accessed 25-February-2020]
- [10] Power BI - Find clarity when you need it most. <https://powerbi.microsoft.com/enus/>. Accessed: 2020-09-24
- [11] KNIME - End to End Data Science. <https://www.knime.com/>. Accessed: 2020-09-19
- [12] RapidMiner Webpage. <https://rapidminer.com/>. Accessed: 2020-09-20
- [13] Orange - Data Mining Fruitful and Fun. <https://orange.biolab.si/>. Accessed: 2020-09-19
- [14] WEKA Webpage. <https://www.cs.waikato.ac.nz/ml/weka/>. Accessed: 2020-09-20
- [15] KEEL - Knowledge Extraction based on Evolutionary Learning. <http://www.keel.es/>. Accessed: 2020-09-19
- [16] Wang XM, Zhang TY, Ma YX, Xia J, and Chen W. A Survey of Visual Analytic Pipelines. *Journal of Computer Science and Technology* 2016 Jul; 31: 787–804. DOI: 10.1007/s11390-016-1663-1
- [17] Keim DA. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics* 2002 Jan; 8:1–8. DOI: 10.1109/2945.981847. Available from: <https://doi.org/10.1109/2945.981847>
- [18] Altalhi AH, Luna JM, Vallejo M, and Ventura S. Evaluation and comparison of open source software suites for data mining and knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2017; 7:e1204

[19] Kotu V and Deshpande B. Predictive analytics and data mining: concepts and practice with rapidminer. Morgan Kaufmann, 2014

[20] Young RR. Recommended requirements gathering practices. *CrossTalk* 2002; 15:9–12

[21] Requirements Gathering: Types and Methods. <https://digitalmarketing.temple.edu/agervasio/2017/07/18/requirements-gathering-typesand-methods/>. Accessed: 2020-11-11

[22] Kazhamiakin R, Pistore M, and Roveri M. A framework for integrating business processes and business requirements. *Proceedings. Eighth IEEE International Enterprise Distributed Object Computing Conference, 2004. EDOC 2004*. IEEE. 2004 :9–20

[23] Maguire M. Methods to support human-centred design. *International Journal Human-Computer Studies* 2001; 55:587–634

[24] Faranello S. Balsamiq wireframes quickstart guide. Packt Publishing Ltd, 2012

[25] Snyder C. Paper prototyping: The fast and easy way to design and refine user interfaces. Morgan Kaufmann, 2003