Innovation Action (IA)

ICT-14-2016-2017

H2020-ICT-2017-1

Enabling procurement data value chains for economic development, demand management, competitive markets and vendor intelligence

THEY BUY FOR YOU

Deliverable 4.3

Data-shape Aware Automatic Storytelling Methods

Date	29/03/2019
Author(s)	Yuchen Zhao (SOTON), Elena Simperl (SOTON)
Dissemination level	Public
Work package	WP4: Interaction Design and Storytelling
Version	Final

Document metadata

Quality assurers and contributors

Quality assurer(s)	Ahmet Soylu (SINTEF), Till C. Lech (SINTEF)
Contributor(s)	

Version history

Date	Version	Description
08/03/2019	0.1	Initial draft for internal review
21/03/2019	0.2	Format improvement
22/03/2019	0.3	Section 2.4 added
25/03/2019	0.4	Comments after internal peer review
28/03/2019	1.0	Final Draft
31/03/2019	Final	Final version for submission

Executive summary

This document presents Deliverable 4.3 "Data-shape Aware Automatic Storytelling Methods" of the TheyBuyForYou (TBFY) project. It records the system design of a data storytelling system, which will be delivered in M27.

This deliverable is part of WP4 "Interaction Design and Storytelling", which contributes to the TBFY project by creating an automatic data storytelling tool, increasing engagement with the TBFY knowledge graph through novel interfaces, and helping the business cases with any user-centric aspects.

This deliverable aims to:

- Provide a brief survey about existing data storytelling systems to inform the design the proposed system.
- Identify the common user journey of data storytelling and the design of data stories.
- Identify the needed functionality of the proposed storytelling system based on the user journey and data story design.
- Set the modules that support the functionality, based on which the actual system will be implemented from M16 to M27.

Table of Contents

Table of Figures

1 Introduction

Data stories are a new type of medium through which people can convey information in an interesting, engaging, and convincing fashion. A data story uses facts (e.g., statistical results and data visualisations) extracted from data to provide insights, support arguments, or call for actions. Creating data stories require both domain knowledge about the topic of the stories and technical skills such as data analysis and data visualisation. In this deliverable, we present the system design of a data-shape aware storytelling tool that aims to help authors who are domain experts or professional journalists but do not have adequate technical skills with creating well-structured data stories. To make data storytelling "data-shape aware", the system will allow authors to inspect the used data in terms of field distribution and uncertainty through visualisations, thereby making them aware of the quality of the used data. It will also be aware of the authors' interests and the transition cost between visualisations. To achieve "automatic storytelling", the system will provide a structured story template and visualisation recommendations to authors, thereby alleviating the authors' burden in storytelling. Both these design considerations aim to address the technical difficulties in data storytelling.

The aim of this deliverable is to provide an overview of how the propose system's functionality is decided and a plan of how the functionality will be implemented by different system modules. It starts with the functions that provided by the existing data storytelling systems and then focuses on the process of data storytelling from users' perspective. The detailed API definitions of the system are not included in this deliverable and will be delivered together with the actual implementation of the system in M27.

The target audience of this deliverable is twofold. Within the TBFY consortium, this deliverable can inform the partners about how they will be able to use the proposed system to create data stories, which can benefit their own business cases that consumes the TBFY knowledge graph. For the audience outside the consortium, this deliverable provides information about the state of the art in data storytelling systems and can indicate the potential challenges in the interaction between storytellers and data, which contribute to the ideation of the future research aiming to address these issues.

This deliverable is related to several other deliverables from different WPs. Within WP4, D4.4 provides a detailed data visualisation guideline, which is helpful for the visualisation design in data storytelling. In D6.1, there are several business cases that can use the proposed system to create data stories that may increase the engagement of their users. The resulting data stories can also contribute to impact report deliverables in WP7.

The rest of this deliverable is organised as six sections. In Section 2, we survey the existing data storytelling tools, and analyse how authors use these tools to create data stories and what components the stories contain. In Section 3, we describe the typical user journey for authors to create data stories with the help of storytelling tools. In Section 4, we describe the functionality needed in different stages of the user journey to help with creating data stories, including loading data, exploring data, structuring stories, visualising data, and annotating stories. In Section 5, we present the different modules of the proposed tool and how they can support the needed functionality. In Section 6, we discuss our future plan for evaluating the proposed system. Finally, in Section 7, we conclude the contributions of this deliverable and discuss future plans.

2 Data storytelling tools

To help us with the system design, first we investigate a number of existing data storytelling tools. We focus on two aspects in our analysis. One is how authors use these tools, which is the process of creating data stories. The other is what components the created data stories have. These two aspects help us design the user journey of our proposed tool, which components should be considered in a data story, and which functionality our tool should have to support the user journey and story components. We analyse both commercial software and research prototypes to understand what are already available and what are currently being studied, based on which we propose the needed functionality of our tool.

2.1 Methodology

Our proposed system aims to help with the storytelling in the whole life cycle of a data story. The life cycle starts from raw data, proceeds through transforming data to visualisations, and ends with combining visualisations with other characters as a resulting story. Thus the purpose of the brief survey in this section is to analyse existing tools that support authors in this whole life cycle and to identify the functionality that our system should provide.

There are two criteria in our selection of existing data storytelling tools. The first criterion is that the selected tool must provide support throughout the whole life cycle. This is to make sure that we can summarise the user journey of data storytelling from how people use the tool. Thus tools that only transform data to visualisations are excluded. The second criterion is that the tool must support data storytelling in general domains, rather than a specific topic. This is to make sure that we can summarise the essential components in the design of data stories. Therefore tools that focus on specific types of visualisations and stories [8, 9] are excluded.

For commercial data visualisation tools and open-source tools, we use Google to search terms "data storytelling tools" and filter the search results according to our criteria discussed above. For research prototypes, we use Google Scholar, ACM Digital Library, and IEEE Xplore Digital Library to search terms "data storytelling" and "narrative visualisation". We have identified the latest survey article on data storytelling by Tong et al.[10]. We filter the research articles from the survey according to our criteria and combine them with additional search results including papers relevant papers published after the survey and a prototypical methodology used by one of our external partners.

2.2 Commercial and open-source tools

We analysed two popular commercial tools, which are Tableau[1] and Microsoft Power BI[2], and one popular open-source tool, which is Jupyter Notebook[3], for data visualisation and storytelling.

2.2.1 Tableau

Tableau is an interactive data visualisation tool. It enables users to load data from different sources, and create data visualisations, dashboards, and stories. Although the main focus of Tableau is business

[1] https://www.tableau.com/
[2] https://powerbi.microsoft.com
[3] https://jupyter.org/

intelligence, its functions are suitable for data storytelling. The typical user journey of using Tableau to create data stories is as follows:

- Connecting: Tableau can connect different types of data sources both locally and remotely. It automatically recognises the types of the fields in the connected data sources and categorise them into dimensions and measures.
- Visualising: Tableau provides easy-to-use functions for data visualisation. A user can simply drag the label of a field into a "column shelf" or a "row shelf" to decide if the field should be visualised as a column value or as a row value. Based on the fields in the shelves, a number of recommendations for data visualisation will be provided to the user. Created visualisations can be saved for storytelling.
- Storytelling: Tableau enables users to create data stories as a series of slides. A data story created by using Tableau has a linear structure. Each of the slides can be edited individually. Users can put texts, visualisations, and annotations in the content of a slide.

A story created through Tableau has three main components:

- Visualisations: A number of visualisations created from data, which can be reused in storytelling.
- Slides: A linear story structure that contains a number of slides.
- Annotations: Additional information such as texts, points, and areas.

2.2.2 Microsoft Power BI

Microsoft Power BI is also an interactive tool that helps users tell stories from data. The functions provided by Microsoft Power BI are similar to those of Tableau. However, there are several differences between the two tools. The user journey of storytelling in Power BI is as follows:

- Getting data: This phase is similar to the *Connecting* phase of Tableau. Users can import data in different formats into a Power BI report and explore the data. Basic field type recognition is provided.

- Visualising and storytelling: Unlike Tableau, Power BI does not have an individual data visualisation phase. Users go through a number of pages of a story and directly create visualisations on each page. The creation of visualisations is similar to that in Tableau. Users first specify a type of visualisation that they want to create and then drag fields into the different dimensions (i.e., x-axis or y-axis) of the visualisation draft. However, unlike Tableau, Power BI does not provide recommendations to users.

Although the user journeys of Tableau and Power BI are slightly different, a story created by Power BI contains the same components as those in Tableau, which are visualisations, slides, and annotations.

2.2.3 Jupyter Notebook

Jupyter Notebook is a web-based platform that allows users to combine code, texts, and visualisations together as a notebook file. The notebook is structured as a series of input sections (e.g., code and data) and output sections (e.g., visualisations generated by code).

As a platform that was mainly designed as an interactive programming environment, Jupyter Notebook does not have a specific procedure for data storytelling. To create a data story through Jupyter Notebook, the process is led by the users' code, including importing necessary libraries, loading data, using libraries to create data visualisations, and exporting visualisations. Thus, to use Jupyter Notebook as a data storytelling tool, the users are expected to have programming skills.

2.3 Research prototypes and frameworks

We also analysed a number of research prototypes and frameworks to understand what kind of new functionality in data storytelling has emerged.

2.3.1 Ellipsis

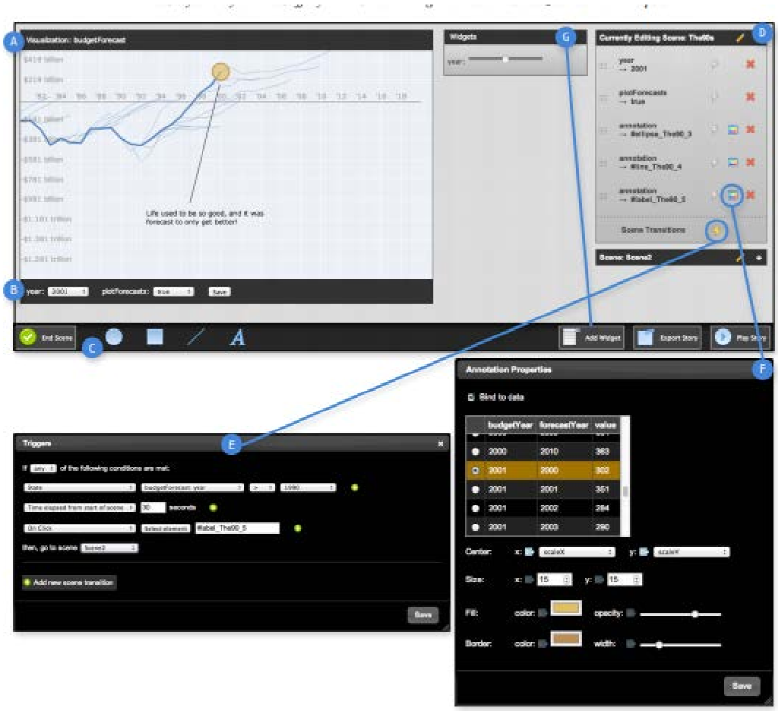

Figure 1. Interface of Ellipsis (Figure from [1])

Ellipsis [1] is a prototypical tool that helps journalists create narrative visualisations without programming. It was designed based on a three-stage user journey:

- Exploring: Exploring datasets to undercover interesting stories in data.
- Drafting: Designing prototypical ways of telling the stories found in exploration.
- Producing: Making the final version of the stories

A data story created through Ellipsis contains four components as follows:

- Visualisations: One of the core components of a data story. Each visualisation component is described by a name-value pair.
- Control widgets: Providing interaction triggers such as click, hover, etc.
- Annotations: Providing textual information along with visualisations.

- Narrative state machine: A number of scenes, each of which contains some visualisations, interactions, and annotations. Scene templates and sub-scenes are provided.

2.3.2 MIT DIVE

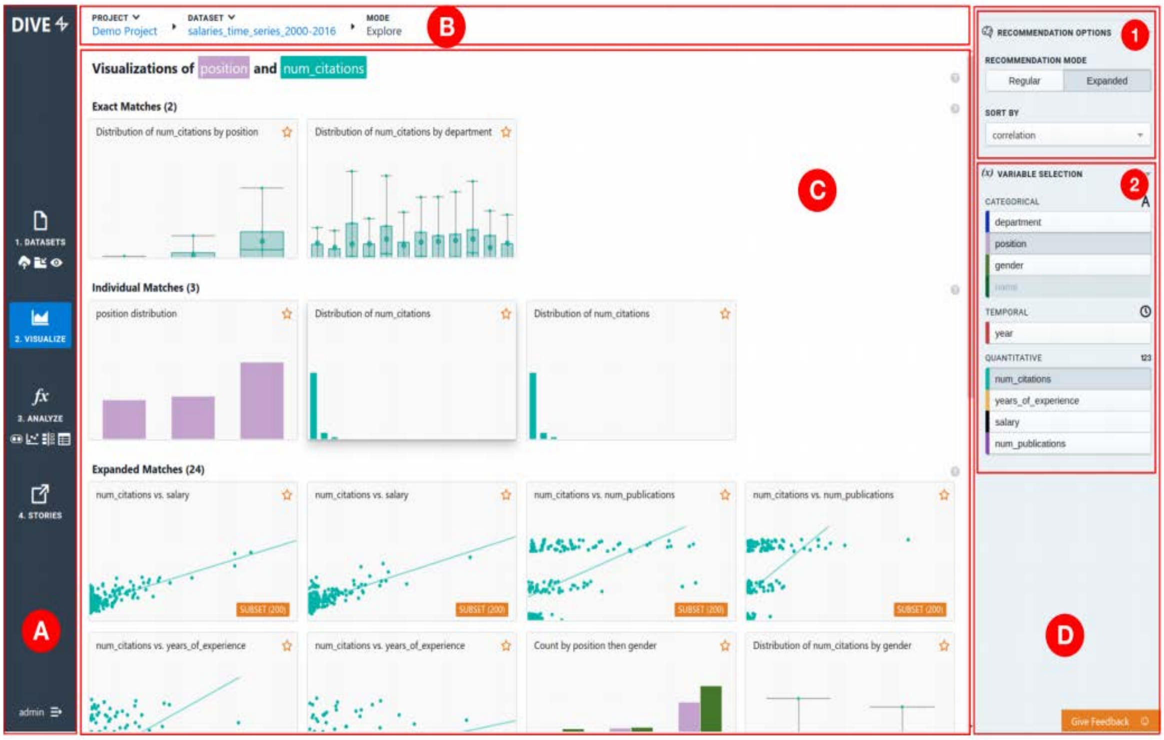

Figure 2. Interface of MIT DIVE (Figure from [2])

MIT DIVE [2] is a web-based platform that provides different functions during data exploration. Authors can use DIVE to conduct data inspection, visualisation, statistical analysis, and storytelling. The user journey in DIVE includes four stages as follows:

- Processing data: Firstly an author uploads a dataset in CSV format. A number of predefined datasets are also available. The second step is to inspect the imported dataset, including field types and the distribution of the values of each field. The field types can be changed.
- Visualising: The first step is to explore a set of recommended visualisations based on the author's interests. The recommendations are sorted. The second step is to select individual visualisations and save them for the final data story.
- Analysing: Aggregation allows the author to count and group data points. Correlation generates correlation matrices that indicate relationships between fields. Comparison allows the author to statistically compare data in different groups. And regression estimates relationships between fields.
- Storytelling: Combining the saved visualisations and analyses together with texts into a story.

A data story made through MIT DIVE has three components:

- Visualisations: Recommended by the system and saved by the author. The recommendations are based on the author's interested fields and ordered by their relevance to the fields. The author explores the recommendations and save the ones that she wants to use in the final story.
- Statistical analyses: Including correlation, aggregation, comparison, and regression. Results can be selected and saved by the author for the final story.
- Linear structure: Composed by a number of titles, headings, descriptions, visualisations, and analyses.

2.3.3 Temporal Summary Images

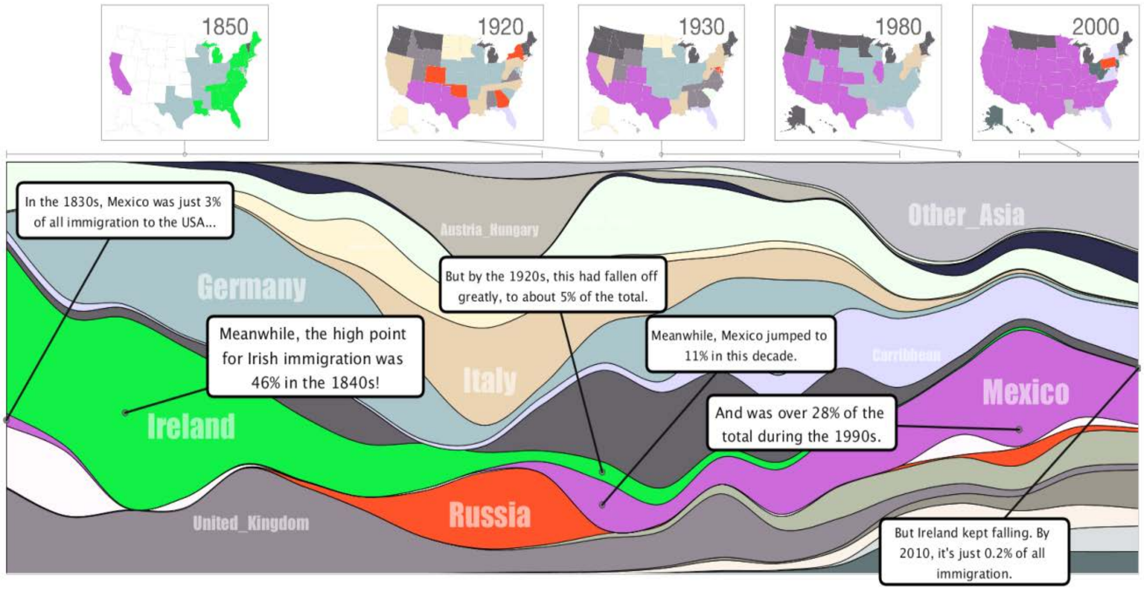

Figure 3. Example of a temporal summary image (Figure from [3])

Temporal summary images (TSIs) [3] is a framework that creates data stories from time-varying data. To create a TSI story, there are five stages as follows:

- Loading: Loading a number of datasets. Apart from the datasets, configuration files, temporal layouts, and pre-defined annotations can also be loaded. An initial temporal layout will be generated based on the loaded data.
- Selecting: The parameters of the story such as temporal layout, algorithms for time step selection, colour palettes, and pre-saved annotations.
- Generating time steps: Using the selected algorithm to generate a series of time steps on which a number of snapshots will be generated.
- Generating snapshots: On each of the generated time step, generating a snapshot of the data at that time step.
- Generating annotations: On each Point of Interest (POI), generating an annotation.

Components of a TSI story:

- A temporal layout: The main component of a TSI story. The layout is a temporal graph (e.g., a stacked graph) generated from the time-varying data.

- A set of time steps: Generated by different algorithms from the time-varying data.
- A set of snapshots: Each snapshot is based on the data at a time step.
- A set of annotations: Textual information at POIs (e.g., point of minimum, point of maximum).

2.3.4 Data Nutrition Label

Unlike normal data stories that use data to create stories about certain topics, the Data Nutrition Label (Label) [7] provides a framework that tells stories about data itself. The aim of Label is to extract useful information from datasets, thereby using this information to help people understand the datasets.

The framework uses several modules to generate labels about datasets from different aspects, such as metadata, data provenance, statistics, and so on. Some labels need high user involvement. For example, users need to manually input the information about metadata and variables. For some other modules, such as statistics, the labels can be automatically generated from pre-designed algorithms. The stories made by the framework are composed by several labels made by different modules, each of which describes the dataset from a specific perspective.

2.3.5 Data Recipe

Similar to Data Nutrition Label, the Data Recipe[4] is also about datasets themselves. As the aim of the Data Nutrition Label framework is to help people understand the quality and bias of datasets to train better machine-learning models, the Data Recipe aims to help journalists understand what is in a dataset and how to use the dataset to tell data stories.

A reporting data recipe is mainly created manually. It is composed by a number of sections including an *Overview* section conveying the aim of the recipe, an *Understanding the data* section presenting the metadata and the methodology used in the recipe, an *Our findings so far* section highlighting the insights and results from the dataset, a *What to look for* section discussing the potential topics for further data stories, and an *Additional data/resources* section pointing to other external sources. A certain level of data journalism skills and domain knowledge are needed to create a data recipe.

2.4 Summary

In Table 1, we summarise the analysed tools in terms of whether programming skills are needed to use the tool, the level of automation of the tool, and the level of awareness provided by the tool. Most of these tools, except Jupyter Notebook, do not require users to have programming skills to make data stories. In terms of the level of automation, some tools (e.g., Tableau and Power BI) automatically produce visualisations and recommend the most suitable one (e.g., Tableau), which makes the storytelling process automatic. But some tools (e.g., Jupyter Notebook, Data Recipe) require users to process data or produce visualisations manually. In terms, of awareness, most of the existing tools only show data tables to users without other information. An exception is MIT DIVE, which informs users about the value distribution of each field.

Our proposed system aims to enable users without programming skills to create data stories. It will provide automatic storytelling by not only providing visualisation generation and recommendations,

[4] https://www.thebureauinvestigates.com/local/open-resources

but also using a structured story template to guide users in their story creation. In addition, it will provide high level awareness on the uncertainty of the data, users' interests, and transition costs.

Table 1. Summary of analysed data storytelling tools and our proposed system

	Tool	Programming skills needed	Level of automation	Level of awareness
Commercial and open-source tools	Tableau	NO	High	Low
	Power BI	NO	High	Low
	Jupyter Notebook	YES	Low	None
Research prototypes and Frameworks	Ellipsis	NO	Medium	Low
	MIT DIVE	NO	High	Medium
	Temporal Summary Images	NO	Low	None
	Data Nutrition Label	NO	Medium	Low
	Data Recipe	NO	Low	Low
	Proposed system	**NO**	**High**	**High**

3 User journeys and story design in storytelling

In this section, we present the user journey of using our proposed tool to create data stories and the design of data stories, based on the analysis in Section 2.

3.1 User journey

Alice is a data journalist who wants to use our tool to make a data story from a public procurement dataset. The first step that Alice needs to do is loading the dataset into the tool. The dataset is a CSV file that contains fields (i.e., columns) describing different procurement information such as tender ID, buyer name, seller name, sector, and so on. Each row in the dataset represents an individual tender.

After the dataset is loaded, Alice can **inspect** it as a table. The tool will automatically detect the types of the fields in the table and present the results next to each field. She can check if the detected results are correct and edit some of them if necessary. She then **explores** the distribution of the values of each field to decide whether a field is of her interests. During the exploration, she can **mark** any fields of her interests and whether a field is a dependent variable or an independent variable.

After inspecting the data and marking interesting fields, Alice needs to **design the story's structure**. The structure provided by the tool combines visualisations and textual information together and connects them in a logical way to support the main argument of her story. She then goes through each part of the structure template and edits the content.

For each part of the story template, Alice **designs visualisations** that fit into that part. The tool will recommend visualisations based on different metrics such as Alice's interests, dependency, uncertainty, and so on. By this means, Alice can simply design visualisations by selecting from the recommendations instead of making from visualisations by herself.

While designing the visualisations in each part of the story, Alice can also use **annotations** to add textual information or emphasise important parts of the visualisations. These include texts, shapes, images, and highlights.

In the whole life cycle of data storytelling, our proposed system will help Alice in different stages. When Alice inspecting and exploring the loaded data, the system will inform her about whether the imported data has any error. It will also present the distribution of the values of each field to help her understand whether a field is good enough or relevant to her story. When she authoring different parts of the story, the system will suggest what actions (e.g., making an argument or using visualisations to support an argument) she should take, thereby making the story more structured. When she generating visualisations from the data, the system will recommend the visualisations that are relevant to her interests and have high quality. By this means, the workload will be alleviated, and the quality and accuracy of the story will be improved.

3.2 Story design

From our analysis in Section 2, we have identified three most important components in the design of data stories, which are story structures, visualisations, and annotations. Our tool will consider these three components in story design.

The structure of a data story is decided by the story's type and the purpose of the author. Most data stories aim to make arguments based on facts and evidence extracted from the used data. This type of story starts with a section that claim an argument. This section must be followed by a series of sections that use facts as evidence to support the initial argument. These facts will eventually be concluded in a final section. By this means, the initial argument can be proved by a final conclusion or a number of sub-conclusions.

In data stories, visualisations are used to reveal important information from the data. The used visualisations are decided by the author's interested fields and the fields' types (e.g., categorical or numerical, discrete or continues, time varying or time independent) and the analysis that the author wants to conduct (e.g., finding trends, comparing entities, exploring geographically). Therefore, to help the author choose visualisations for the story, our tool will provide visualisation recommendations. These recommendations are based on the interests of the author, the dependency between fields, the uncertainty of the visualised data, and the transition cost between different visualisations.

Apart from visualisations, another important component of a data story is annotations. Visualisations cannot demonstrate all the information in data by themselves. Therefore annotations are needed to present texts, highlights, shapes, or images. These different annotations service different purposes and are associated with different targets. For example, texts are used as titles, headings, and descriptions while highlights are used to emphasise the most important part of a visualisation.

4 Functionality requirements

In this section we describe the required functionality of our tool, based on the user journey and story design discussed in Section 3.

4.1 Loading data

The tool will support two types of data:

- Pre-loaded datasets: Some pre-loaded demo datasets will be provided with the tool. These datasets can be used by authors as tutorials to familiarise themselves with the functionality of the tool. The datasets will be chosen from existing public procurement data and will be modified to make sure that the authors can use these datasets to experience most of the functions of the tool and can make interesting stories from the data.
- Offline datasets: The tool will support datasets in CSV format. Basic functions such as file error detection will be supported.

4.2 Inspecting and marking

To enable authors to inspect the loaded data and mark the fields of their interests, the tool will support the following functions:

- Field type detection: For each field in the loaded dataset, the tool will use some heuristics to detect its type. Commonly used types such as categories, numbers, dates, and missing values will be identified.
- Field type modification: The tool will show the detected types of fields to the author and will allow the author to modify any incorrect types.
- Field distribution visualisation: The tool will use histograms to show the distribution of the values of each filed. This is to help the author understand the high-level information of the dataset and each field. Based on the histograms, the author can gain insights about whether two fields are related or whether a field is useful to contribute to a story.
- Field dependency marking: The author can mark two fields as "dependency" if she thinks that one field may be dependent on the other one. This information will be used for visualisation recommendations.
- Field of interest marking: The author can mark the fields that she thinks are useful for her story, from looking at their distributions, or from her knowledge about the topic. This information will be used for visualisation recommendations.

4.3 Story structure design

Our tool will provide a structured template for data storytelling. The template contains three basic logical building blocks: *Claim*, *Fact*, and *Conclusion*.

Claim, Fact, Conclusion (CFO) template [4]: The template will guide the author to make a story that can convey arguments in a structured and logical way. The story starts with a *Claim* section. The content of the *Claim* section is mainly textual and does not have evidence as proof. To support the initial claim, a number of *Fact* sections will follow as evidence. These sections contain visualisations made from the imported data. Each *Fact* section proves a point that contributes to a *Conclusion* section. Therefore a number of *Conclusion* sections can be supported by the *Fact* sections. The overall argument of the story, which is presented as the initial claim, will be supported by the *Conclusion* sections. Figure 4 shows the procedure of the authors using our tool to tell a data story under the CFO structure.

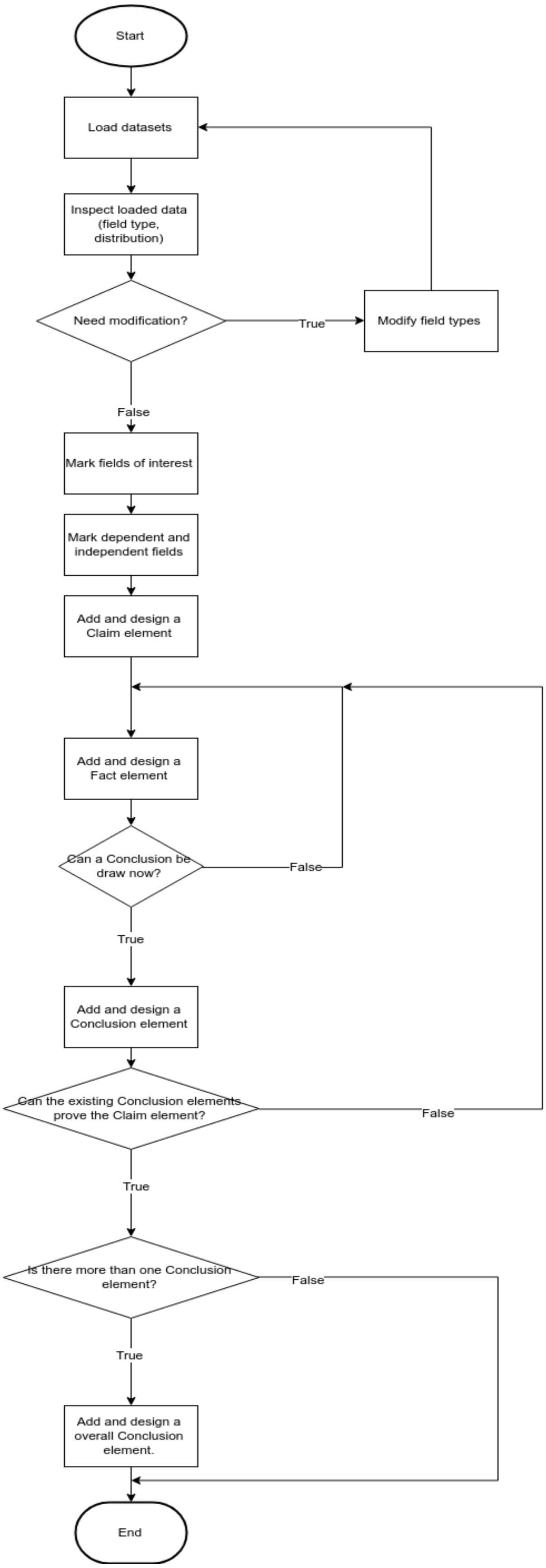
Start
Load datasets
Inspect loaded data (field type, distribution)
Need modification?
True
Modify field types
False
Mark fields of interest
Mark dependent and independent fields
Add and design a Claim element
Add and design a Fact element
Can a Conclusion be draw now?
False
True
Add and design a Conclusion element
Can the existing Conclusion elements prove the Claim element?
False
True
Is there more than one Conclusion element?
False
True
Add and design a overall Conclusion element.
End

Figure 4. Procedure of data storytelling through the CFO structure template

4.4 Visualisation design

In *Fact* sections, the tool makes recommendations for visualisations to help the author create her data story more easily. These recommendations contain the most commonly used visualisations for different purposes and are based on the interests of the author, the dependency of fields, the uncertainty of the data, and the transition cost from one visualisation to another.

- Basic types of visualisations: The tool will provide basic types of visualisations including bar charts, line charts, and tree maps.
- Interest-aware recommendation: A data story is based on an author’s interests. From the same imported dataset, the output story may vary when the author’s interests change. Such interests are collected in the inspection stage and are used to recommend visualisations. For example, the author can select two fields as of her interests because she thinks they may be correlated. Or she can select a field as of her interests because she believes that there may be a trend in this field. Specifically, a number of visualisations will first be made based on the types of the fields in the dataset. Fields of the author’s interests and relevant visualisations will be recommended first. By this means, the author can easily find the visualisations most relevant to her story.
- Dependency-restricted recommendation: For all the fields in the imported dataset, there are dependencies between them. In the inspection stage, the author uses her domain knowledge to mark the dependencies. Such dependencies will be used to refine the generated visualisations. Visualisations that more fit into the dependency restrictions will be considered more suitable for the story and will be recommended to the author.
- Uncertainty-aware recommendation: Depending on the quality of the used data, there might be uncertainty in a visualisation. For example, when using a bar chart to present average values, the uncertainty comes from the size of the samples used to calculate the average. Our tools will consider the uncertainty of the results in visualisations and recommend visualisations with lower uncertainty than others as they are more likely to reveal accuracy and high quality insights.
- Transition-cost-aware recommendation: As a data story normally contains more than one individual visualisation, the audience of the story needs to go through one visualisation after another. In this sequential story structure, the order of the visualisations affects the reading efficiency of the audience, as there is transition cost between different types of visualisations [6]. The tool will consider the transition cost between the visualisation of the previous section and the possible recommendations of the current section. Visualisations that have lower transition cost than others will be recommended first.

4.5 Annotation design [5]

Apart from visualisations, the tool will also provide annotations to help create data stories. These annotations are in different forms and can be used for different targets in the stories. Ren et al. [5] have discussed the classification of annotations based on these two dimensions (i.e., form and target). In the next two sub-section, we briefly introduce this classification, based on which the annotation functionality of our system will be implemented.

4.5.1 Multiple forms of annotation

Our system will support annotations in four different forms as follows:

- Text: Textual annotations can be used as titles, headings, descriptions and so on in data stories. They are necessary in *Claim* sections and *Conclusion* sections. In *Fact* sections, they can also provide additional information that is hard to be presented by solely using visualisations.
- Shapes: Shapes, including lines and polygons, can be used to present directions and areas. They are useful for showing trends and groups in data visualisations.
- Highlights: Highlights such as colours and sizes can be used to emphasise or de-emphasise entities in visualisations. They can help to make audience focus on certain areas and pay their attention to the most important information.
- Images: Images can be used to show the information that cannot be shown by visualisations or texts, including logos, photos, icons, and so on.

4.5.2 Multiple targets of annotation

Our system will support annotations related to four different types of targets as follows:

- Data item, set, and series targets: These targets are related to the imported data. For instance, a single data point on the line of a line chart is a data item target for annotation. A series of continuous data points on the line are series targets. Discrete data points are treated as set targets.
- Coordinate space targets: These targets are based on the axes of the coordinate space of a chart. For instance, horizontal lines can be used to represent certain threshold values of a y-axis or to mark certain areas between two different values.
- Chart element targets: These targets including the chart elements that are not generated from data, including titles, headings, labels, and so on.
- Prior annotation: Existing annotations can be used as the targets of new annotations.

5 System design

In this section, we present the system modules of our proposed tool. Each of them supports a set of functionality described in Section 4. The diagram of these modules is shown in Figure 5.

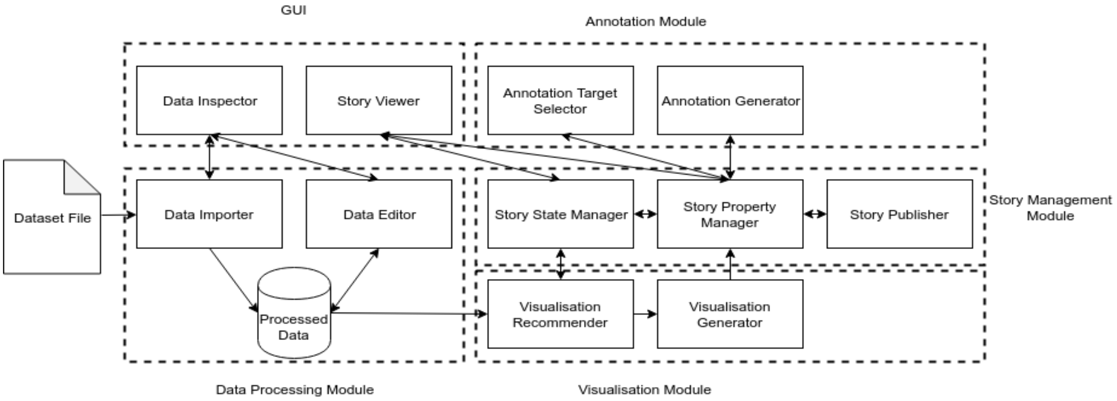

Figure 5. Diagram of different modules in the system

5.1 Graphic User Interface (GUI)

The GUI is the front-end interface of the tool. It provides two visual interfaces to the user. One interface, which is a data inspector, allows the user to inspect the imported data and the distributions of the fields of the data. The user can also select the fields and edit through this interface. The other interface, which is a story viewer, allows the user to browse the created story, modify the content of each section of the story, and select the recommended visualisations. The wireframes of the data inspector and the story viewer are shown in Figure 6 and Figure 7, respectively.

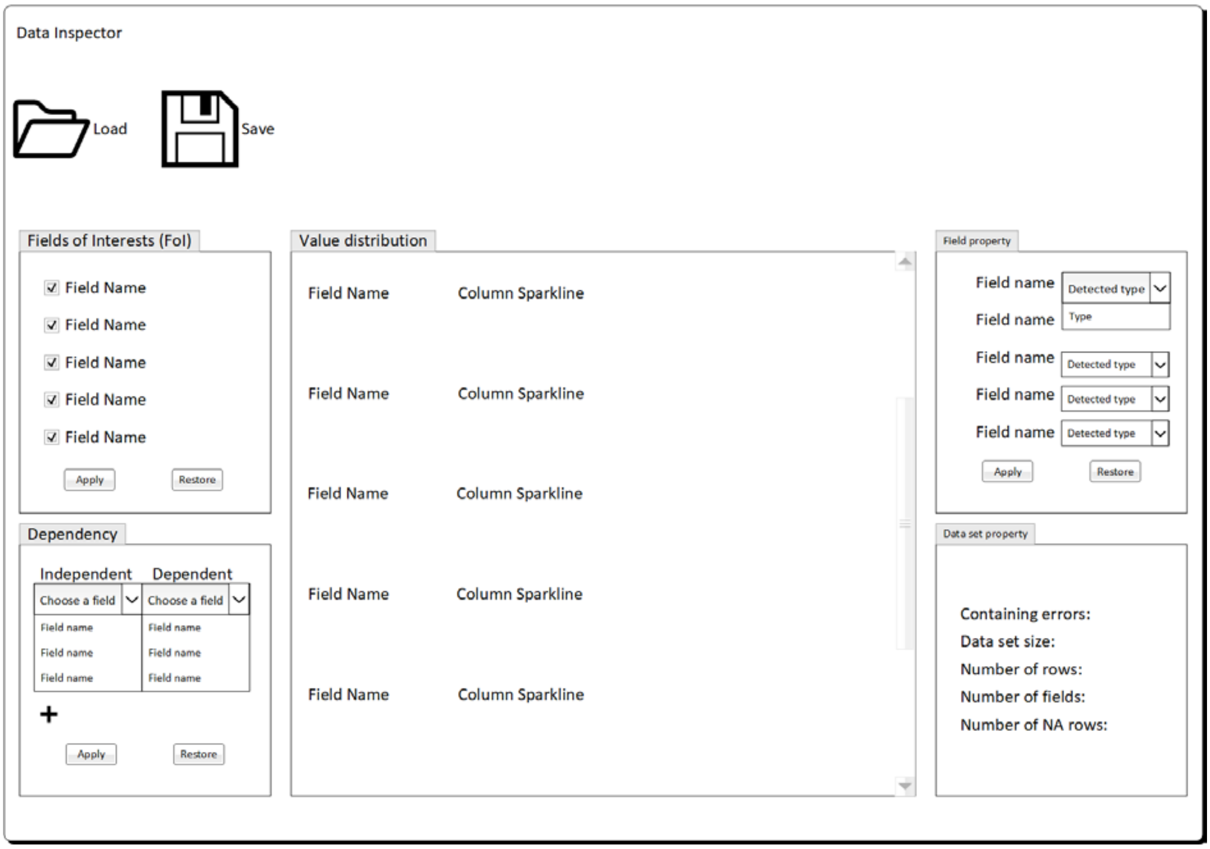

Figure 6. Wireframe of the data inspector interface

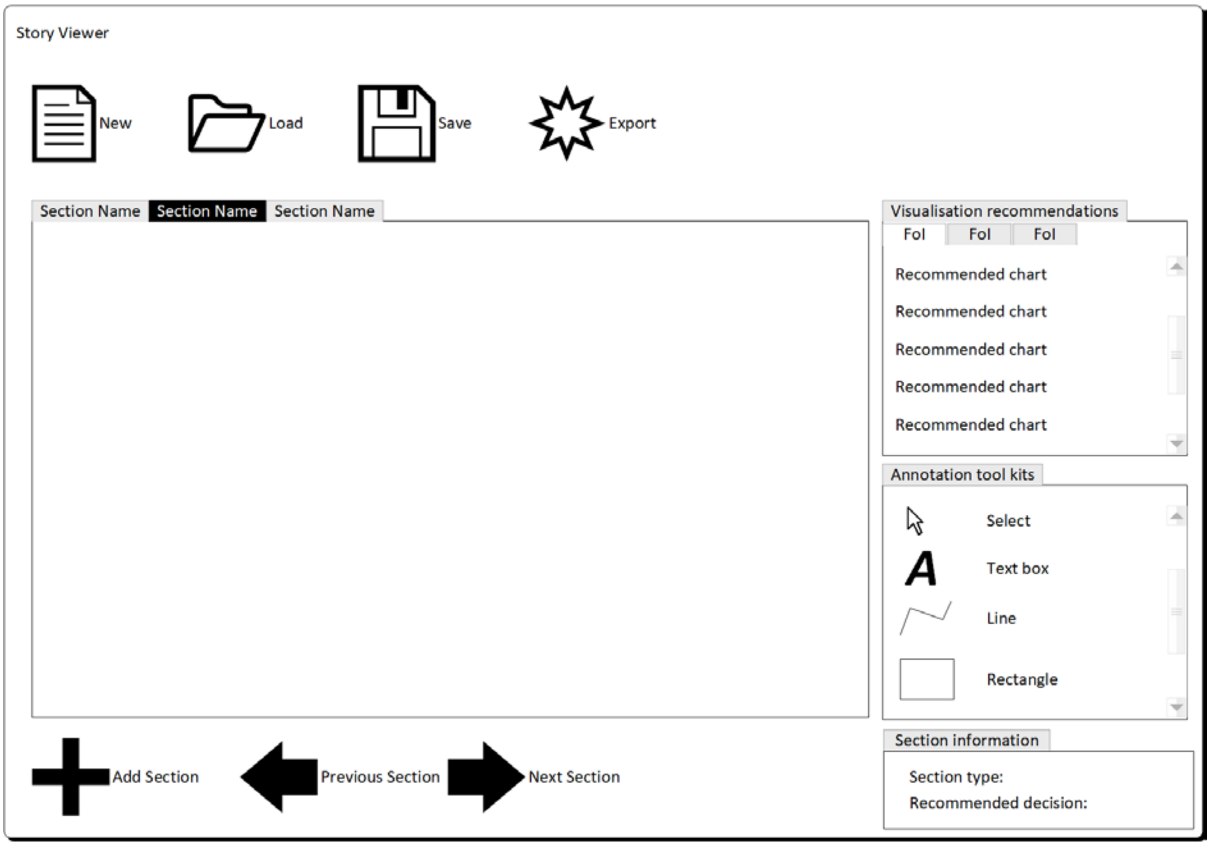

Figure 7. Wireframe of the story viewer interface

5.2 Data Processing Module

The Data Processing Module contains a data importer and a data editor. The data importer works at the backend of the tool and loads the user's specified dataset into the workplace as processed data. The data editor receives instructions from the user through the data inspector in the GUI and apply corresponding modifications on the processed dataset.

5.3 Story Management Module

The Story Management Module is one of the main modules of the tool. It manages the states and components of a data story. The story state manager records the structure of the created story and its current state (i.e., the section that the user works on). It communicates with the story viewer in the GUI and allows the user to navigate between different sections of the story. The changes of the state of story affects the recommendations for visualisations. The story property manager records the components of each section of the story. These components include used data, visualisations, annotations, and layout. By combining the story structure and components, the story publisher can export the story in different formats.

5.4 Visualisation Module

Another main module is the Visualisation Module. This module contains a visualisation recommender and a visualisation generator. The recommender analyses the user's chosen data, interested fields, field dependencies, uncertainty from the data, and the transition cost from previously selected visualisations, to make recommendations to the user for the current state. The user can see these recommendations through the story viewer in GUI and select which one of them she wants to use. The selected recommendation will be created by the visualisation generator and be managed by the story components manager.

5.5 Annotation Module

The Annotation Module has two components that allows the user to select the targets of annotation and to create different types of annotations. The annotation target selector provides interactions between the user and the components of the story. The user can select data items, coordinates space targets, chart elements, and previous annotations as targets for annotations. Based on the selection, the user can choose which type of annotation she wants to create through the annotation generator.

6 Evaluation

In this section, we present our plan for evaluating the proposed system. The evaluation will be user-centric, which means that the system will be used by real users and their feedback will be recorded and be considered to revise the system. We are interested in the feedback from two groups of users, which are the project partners and journalists.

6.1 Evaluation within TBFY

As part of WP4 of the TBFY project, the system aims to help the project partners with their business cases by creating data stories from the TBFY knowledge graph. The system will assist people who have knowledge in public procurement but do not have adequate technical skills in data storytelling. Therefore we would like to ask the project partners to use the system and investigate whether the system can help them create data stories around public procurement. We are interested in their general user perceptions such as how easy the system is to use and how helpful it is in data storytelling. The feedback from the project partners will be adopted to revise the design of the existing functions.

6.2 Evaluation outside TBFY

We are also interested in evaluating the system with the help from data journalists to find out whether it is helpful for data storytelling in other areas. Data journalists have not only domain knowledge in certain areas, but also data storytelling skills. We would like to know whether they think that the functions provided by the system are adequate and what other functions they suggest. Their feedback will be considered for additional improvements on the system.

7 Conclusion

This deliverable has presented the design of a data storytelling system. As part of WP4, this deliverable has analysed how users create data stories, which functionality is needed, and which modules are responsible of supporting the functionality. The project partners can also use this deliverable to understand how their business cases can benefit from the proposed system.

The future plan for this work includes three phases. First, we will be refining the system design in a more detailed level, including API definition and testing plan. Then a minimum viable product (MVP) will be implemented, both as a demo to demonstrate the functionality and as a prototype to be evaluated by users. Finally we will be iterate and revise the design based on the feedback from the evaluation. The final version of the system will be delivered in M27.

8 References

[1] A. Satyanarayan and J. Heer, “Authoring Narrative Visualizations with Ellipsis,” *Computer Graphics Forum,* vol. 33, no. 3, pp. 361-370, 2014.

[2] K. Hu, D. Orghian and C. Hidalgo, “DIVE: A Mixed-Initiative System Supporting Integrated Data Exploration Workflows,” in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, Houston, TX, USA, 2018.

[3] C. Bryan, K.-L. Ma and J. Woodring, “Temporal Summary Images: An Approach to Narrative Visualization via Interactive Annotation Generation and Placement,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 23, no. 1, pp. 511-520, 2017.

[4] R. Kosara, “An Argument Structure for Data Stories,” in *Short Paper Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization (EuroVis)*, 2017.

[5] D. Ren, M. Brehmer, B. Lee, T. Höllerer and E. K. Choe, “Chartaccent: Annotation for Data-Driven Storytelling,” in *Proceedings of the 2017 IEEE Pacific Visualization Symposium (PacificVis)*, Seoul, Korea, 2017.

[6] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher and E. Adar, “A Deeper Understanding of Sequence in Narrative Visualization,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 19, no. 12, pp. 2406-2415, 2013.

[7] S. Holland, A. Hosny, S. Newman, J. Joseph and K. Chmielinski, “The Dataset Nutrition Label: A Framework to Drive Higher Data Quality Standards,” *arXiv preprint arXiv:1805.03677,* 2018.

[8] J. Hullman, N. Diakopoulos and E. Adar, “Contextifier: Automatic Generation of Annotated Stock Visualizations,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 2013.

[9] T. Gao, J. Hullman, E. Adar, B. Hecht and N. Diakopoulos, “NewsViews: An Automated Pipeline for Creating Custom Geovisualizations for News,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Toronto, Ontario, Canada, 2014.

[10] C. Tong, R. Roberts, R. Borgo, S. Walton, R. Laramee, K. Wegba, A. Lu, Y. Wang, H. Qu, Q. Luo and X. Ma , “Storytelling and Visualization: An Extended Survey,” *Information,* vol. 9, no. 3, p. 65, 2018.