


AUTHOR QUERY FORM

 ELSEVIER	Book: Perspectives on Data Science for Software Engineering Chapter: 00015	Please e-mail your responses and any corrections to: E-mail: M.NATARAJAN@elsevier.com
---	---	--

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult <http://www.elsevier.com/artworkinstructions>.

We were unable to process your file(s) fully electronically and have proceeded by

<input type="checkbox"/>	Scanning (parts of) your article	<input type="checkbox"/>	Rekeying (parts of) your article	<input type="checkbox"/>	Scanning the artwork
--------------------------	-------------------------------------	--------------------------	-------------------------------------	--------------------------	----------------------

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the '[Q](#)' link to go to the location in the proof.

Location in Chapter	Query / Remark: click on the Q link to go Please insert your reply or correction at the corresponding line in the proof
Q1	Please provide caption.
Q2	Please provide a caption.
Q3	Please provide a caption.
Q4	The first paragraph has been treated as the abstract. Please check and confirm.
Q5	Please check if the keywords suggested by the editor are appropriate.
Q6	Please check if the insertion of Figs. 1–3 citation is OK, or suggest an alternative location if necessary.

<u>Q7</u>	Please provide volume number and page range for Ref. [1].		
<u>Q8</u>	Please provide publisher location Ref. [3].		
	<table border="1"><tr><td>Please check this box or indicate your approval if you have no corrections to make to the PDF file</td><td></td></tr></table>	Please check this box or indicate your approval if you have no corrections to make to the PDF file	
Please check this box or indicate your approval if you have no corrections to make to the PDF file			

Thank you for your assistance.

c0075

Visual analytics for software engineering data

Z. Hou, H. Zhang, H. Zhang, D. Zhang

Microsoft Research, Beijing, China

p0010 Many data analysis techniques have been widely used in practice to enable software practitioners to perform data exploration and analysis in order to obtain insightful and actionable knowledge for real tasks around software and services [1]. Although these techniques are effective, they require substantial knowledge and skills and are typically performed by “data scientists.” Ordinary users (such as programmers, marketing personnel, service operators, managers, etc.) may find it difficult to apply these techniques to quickly explore the data and obtain insights by themselves. For example, users may have to learn SQL queries in order to extract data from a database, and to learn statistical techniques to analyze the data. Users often need to write programs to analyze data extracted from an Excel/text file. These techniques may increase the cost of data exploration for ordinary users and create high barriers to entry.

p0015 Many methods can be designed to democratize data analysis in software engineering. We advocate for incorporating visual analytics techniques into the analysis of software engineering data. Visual analytics is “the science of analytical reasoning facilitated by interactive visual interfaces” [2,3]. It focuses on the interactive exploration and manipulation of the data. Using visual analytics techniques, a user can perform several successive queries and view the data in a variety of formats before satisfactorily identifying the data in which they are most interested.

p0020 The Software Analytics group at Microsoft Research Asia developed a visual analytics tool called MetroEyes. MetroEyes can import data from external sources (such as Excel files or SQL databases) automatically. Data are represented as visual objects, such as a slice in a pie chart, a bar in a bar chart, or a series legend in a line chart, etc. MetroEyes provides an interactive graphical interface that enables users to directly click/touch/move all these objects. The visual objects can also be composed to form a new chart. MetroEyes is able to interpret the intentions of the visual operations, extract corresponding data from the data source, and create a new chart.

p0025 For example, assume that we have an Excel spreadsheet which contains the app sales data of a multinational software corporation. The corporation has three product teams (code named TeamA, TeamB, and TeamC) developing Game and Education Apps. The data includes yearly app sales in different counties, along with the detailed

78 Visual analytics for software engineering data

sales of different teams and categories. There are five columns (“Year,” “Team,” “Category,” “Country,” “Sales”) in the spreadsheet. Some sample records are shown in the following table.

Year	Team	Category	Country	Sales (M)
2010	TeamA	Game	United States	0.5
2010	TeamC	Game	China	0.5
2010	TeamA	Education	United States	0.1
2011	TeamC	Education	China	0.4
2011	TeamB	Game	United States	0.3
2011	TeamB	Education	China	0.2

p0030 In MetroEyes, users can perform data exploration through direct manipulation of the visual objects. Say the users want to explore the sales of TeamA. As illustrated in Fig. 1, users can directly select the TeamA bar from the bar chart representing the contribution of each team to Sales, and drag and drop it into the canvas. The tool can then extract from the data source the app sales data contributed by TeamA, and display it in a new bar chart. This data operation selects a dimension value (Team = TeamA) and finds out its app sales, which is equivalent to the SQL query: SELECT Sales, Team FROM AppSales WHERE Team = “TeamA.” Note that each bar in the bar chart is a visual object, which can be touched and moved around. Furthermore, each bar represents the percentage of sales each team contributes (eg, the TeamA bar indicates the percentage of sales of TeamA). The visual operations over the object have semantic meanings and correspond to certain data operations.

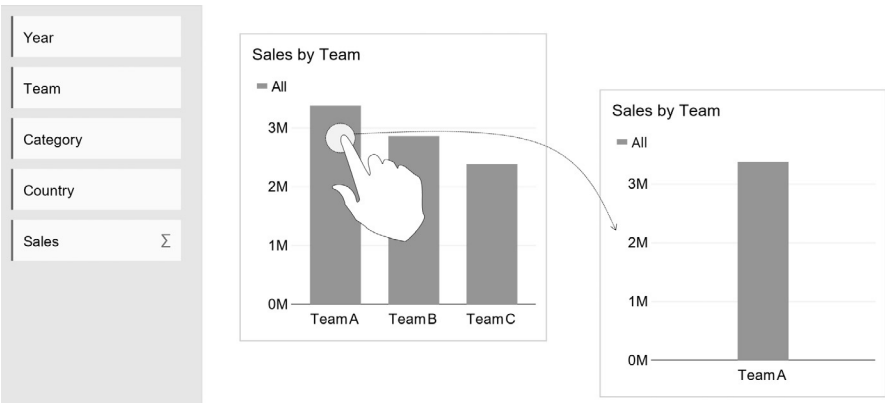


FIG. 1

f0010

Q1

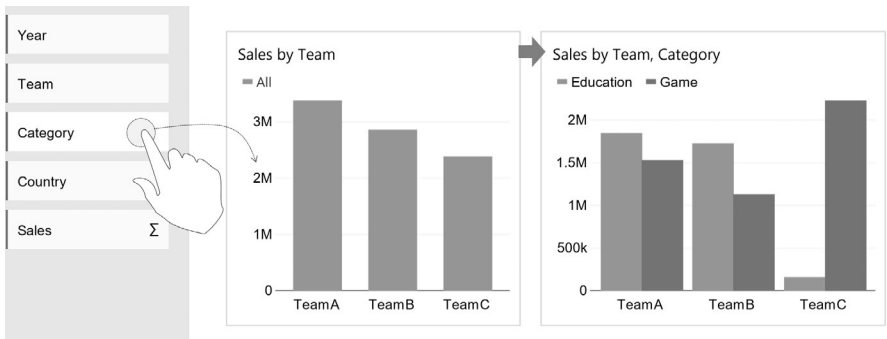


FIG. 2

f0015

Q2

p0035 As another example, say users want to explore the sales data by Team and Category. As illustrated in Fig. 2, users first select the Team dimension and drop it into the canvas. MetroEyes extracts the team data from the data source, and displays a chart that contains the app sales data broken down by Team. Users can then select the Category dimension and drop it into the chart. Finally, MetroEyes displays a chart that shows the Team's sale data, broken down by category. This data operation explores data along multiple dimensions (in this case, the Team and Category dimensions), which is equivalent to the SQL query: `SELECT Sales, Team, Category FROM AppSales.`

p0040 MetroEyes also enables changes from one data visualization format to another, and supports different types of data exploration tasks such as filtering and sorting. For example, data can be sorted through the use of gestures, which is illustrated in Fig. 3.

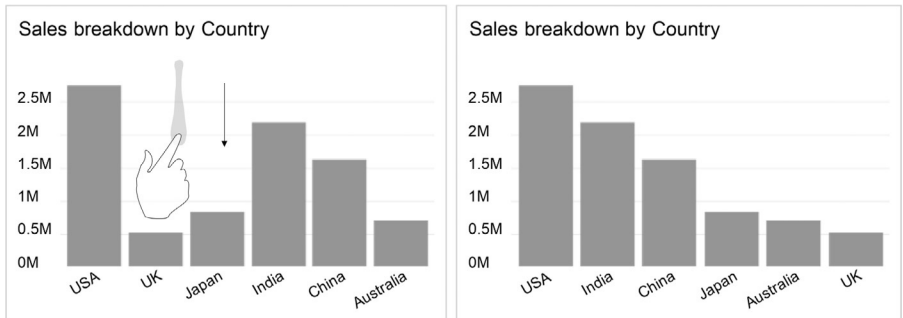


FIG. 3

f0020



Q3

80 Visual analytics for software engineering data

p0045 When using MetroEyes to explore SE data, users do not need to write SQL queries or programs by themselves, or understand sophisticated mathematical concepts. What they need to do is to decide what data they want, and simply conduct direct manipulation over visual objects to obtain the data. Although the visual operations are very simple to perform, they are able to express rich semantics for data operations. Furthermore, users are able to view the graphical representation of the data throughout the entire exploration process, which provides a much more intuitive understanding of the underlying data.

p0050 We believe that data analysis in software engineering should incorporate techniques from visual analytics, in order to help ordinary users analyze and understand their data. In this way, we lower the barrier to entry and reduce the cost of data exploration for ordinary users. MetroEyes has been used internally by Microsoft teams to perform various analytical tasks. We have successfully transferred the main concepts and experiences of MetroEyes to the Microsoft Power BI product (<http://www.powerbi.com>), which was officially released in 2015.

REFERENCES

- [1] Zhang D, Han S, Dang Y, Lou J, Zhang H, Xie T. Software analytics in practice. IEEE Softw 2013. 
- [2] Kielman J, Thomas J, Guest editors. Special issue: foundations and frontiers of visual analytics. Inf Vis 2009;8(4):239–314.
- [3] Thomas J, Cook K. Illuminating the path: research and development agenda for visual analytics. IEEE-Press; 2005. 

Non-Print Items

Abstract

Many data analysis techniques have been widely used in practice to enable software practitioners to perform data exploration and analysis in order to obtain insightful and actionable knowledge for real tasks around software and services. Although these techniques are effective, they require substantial knowledge and skills and are typically performed by “data scientists.” Ordinary users (such as programmers, marketing personnel, service operators, managers, etc.) may find it difficult to apply these techniques to quickly explore the data and obtain insights by themselves. For example, users may have to learn SQL queries in order to extract data from a database, and to learn statistical techniques to analyze the data. Users often need to write programs to analyze data extracted from an Excel/text file. These techniques may increase the cost of data exploration for ordinary users and create high barriers to entry. [\[Q4\]](#)

Keywords: Data analysis techniques, Visual analytics techniques, Interactive visual interfaces, [\[Q5\]](#) MetroEyes, Data exploration tasks