Panoramic View for Visual Analysis of Large-scale Activity Data

Kazuo Misue† and Seiya Yazaki

University of Tsukuba, Tsukuba, Japan
†misue@cs.tsukuba.ac.jp

Abstract. Understanding the activities in large-scale organizations such as big companies is very important. A challenge in the information visualization field is how to combine a representation of the global structure of an organization with representations of each activity. We developed a representation technique to provide a panoramic view of such activities. The representation embeds charts expressing activities into cells of a treemap. By using this representation, both quantitative and temporal aspects of activities can be seen simultaneously. We also developed an analysis tool called "Series at a Glance," which provides functions to manipulate the representation. The tool helps in the analysis of tens of thousands of activities by providing useful visual information.

Key words: panoramic view, visualization of activity data, treemap, Gantt chart, issue tracking system

1 Introduction

In a big company, more than several thousand projects are being run every year. The progress of each project is often managed by referring numeric data such as its costs and profit. For the management of projects, there is a lot of detail data as important as such the summarized numerical one. While upper management should take responsibility for more projects, it is difficult for them to pay attention to detail of every project. Therefore, it is useful if they can grasp the progress of a lot of projects simultaneously.

Our challenge is to help to understand the activities in a large-scale organization such as a big company. Many existing management tools and analysis tools can help in determining particular characteristics of these activities. However, these tools have been designed for well-known characteristics, so we cannot apply them for revealing unfamiliar characteristics in many activities. To determine unknown characteristics, a wide-ranging observation of activities is essential, and therefore the panoramic view of activities should be useful. Our technical challenge is to develop a panoramic view of activities to help users to fully understand them.

We adapted tickets of the issue tracking system as activity target data. We developed a visualization technique for the tickets. The technique gives a panoramic

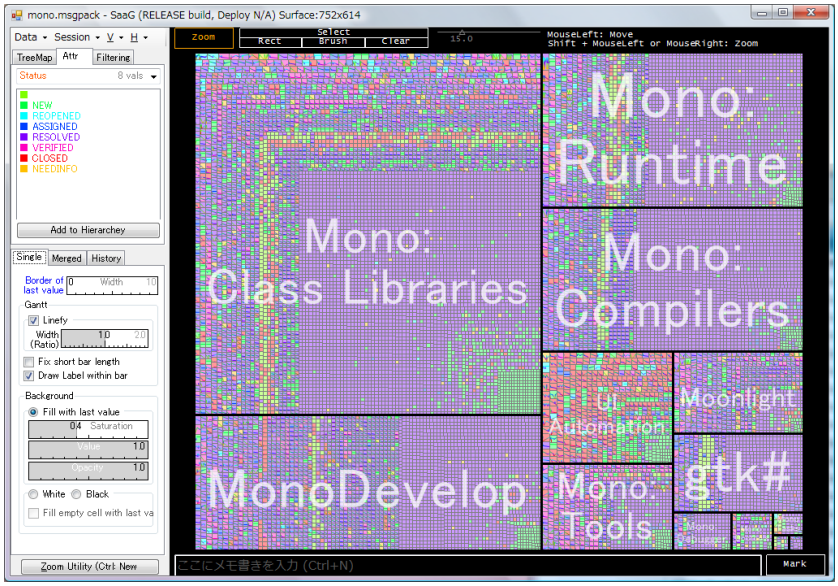

Fig. 1. Screen shot of Series at a Glance, in which about 20,000 tickets are displayed as a treemap. All tickets are categorized by Product and are drawn as polyline charts (see Section 4.2) in Tiling mode (see Section 4.3).

view of the progress of information from many tickets at the same time. Our contributions are as follows:

Hierarchical Representation + Time-line Representation The visual representation we developed expresses the hierarchical structures of ticket groups and the temporal information of each ticket. The representation enables us to observe temporal information of a ticket and ticket groups while being aware of the global structure of tickets.

Visual Analysis Tool for Many Activities We developed a tool that shows tens of thousands of tickets in a window (see Figure 1). The tool provides functions that are required to perform visual analysis processes. Users may change the hierarchical structure of the ticket group, attributes to be displayed, etc.

2 Activities and Their Analysis

We roughly formulated a unit of activity in an organization as follows: A unit has some attributes, and the values of the attributes may change with time. For example, a project can be a unit and the section, products, costs, profits, and status of the project can be attributes. Attributes and the values of the attributes are varied depending on companies.

2.1 Tickets in the Issue Tracking System

Tickets are activity data managed by the issue tracking system (ITS). Although it is difficult to get actual data for specific companies, it is relatively easy to get tickets for open-source software (OSS) projects. A ticket corresponds to a task within the project and records the history of the task. Tickets have attributes such as *Status*, *Assigned to*, *Product*, etc. Attribute values are updated through a project management tool when the task corresponding to the ticket has progressed. For example, the attribute Status may have values of *New*, *Assigned*, *Resolved*, *Verified*, *Closed*, etc. Attributes, the values of the attributes, and meanings of the values are also varied according to projects and organizations. The total number of tickets may exceed 10,000. This is similar to activities in a big company.

2.2 Analysis of Tickets

One goal of our analysis is to find unknown but useful characteristics of some groups of activities. We should observe tickets from various angles to find some useful knowledge in situations in which our interest has never fixed on some specific aspects. One popular approach to achieving this goal is to perform analytical processes along visual information-seeking mantra[1].

Overview First, we should get an overview of all tickets or an entire group of tickets, that is, a panoramic view of the activities. The following two aspects are essential in the overview:

1. the number of tickets and
2. time changes of the attribute values of the tickets.

We can divide tickets into groups according to their attributes. Iteratively dividing tickets into groups can allow construction of various hierarchical structures by the order of attributes of interest. It is useful if the overview of the tickets is based on the hierarchical structure. We can look at the entire group of tickets with interesting attributes as clues.

Zoom Only the tickets in some groups should be enlarged in the window. A group is a set of one or more tickets with specific attribute values. Focusing on a group of tickets means paying our attention to tickets with some specific attribute values.

Filter Tickets with specific attribute values should be excluded. Excluding completed tickets or tickets we are not interested in makes it easy to concentrate on valid tickets that may include some useful characteristics.

Detail on Demand One or several tickets are displayed in detail. About a specified attribute, it is desirable that we can understand changes of the values.

3 Related Work

When we look at our research from the viewpoint of objectives of our ticket analysis, its purpose has something in common with the analysis of software development projects. Therefore, we see our challenge as part of a visualization of the activities in such projects. If a group of tickets is regarded as time-series data, our technique can be considered as one of the visualization techniques of large-scale time-series data.

3.1 Visualization of Software Development

There is a considerable amount of research on visualization of the activities of software development projects that support users to obtain knowledge from them[2]. Ball et al. and Froehlich et al. present visual representations of the history of source programs extracted from a repository[3, 4]. We suppose that we should analyze human activities as well as the history of source programs.

MDS-Views[5] applies multidimensional scaling to data extracted from CVS[1] and Bugzilla[2] and shows relationships among elements in a project as a node-link diagram. Social Health Overview[6] is a tool for evaluating the soundness of activities of a project. It expresses tickets and their attributes extracted from Bugzilla as dots with colors.

Software evolution storylines[7] and code_swarm[8] support observation of changes within a development community. They show the time change of individual contributions or a contribution portion to a project.

Most of the existing tools are unsuitable for observing activity data from varied viewpoints because they only cover some limited viewpoints of analysis and use special measures and simplification based on these viewpoints.

3.2 Visualization of Time-series Data

A lot of research has been conducted on visualization of time-series data. Aigner et al. have developed a systematic view on the diversity of methods for visualizing time-oriented data[9].

To show a large amount of data in a view, it is necessary to increase space efficiency. As examples of techniques considering space efficiency, Reijner developed Horizon Graph[10], Heer et al. improved it[11], and Krstajić et al. proposed CloudLines[12]. Chromograms by Wattenberg et al. [13] can be regarded as one of these techniques. They have devised color mapping to acquire information from series data efficiently.

Although these techniques are equipped with some outstanding feature, we need some more effort to embed tens of thousands of activities to a limited screen area.

[1] http://www.nongnu.org/cvs/
[2] http://www.bugzilla.org/

4 Visual Representation for Large-scale Activities

Our requirements to realize an overview of all tickets are (a) the number of tickets and (b) time changes of the attribute values of the tickets. The number of tickets can be tens of thousands. The area for each ticket should be small to get an overview of all tickets. The set of tickets comprises a hierarchical structure as a global structure and every ticket has a temporal structure as a local structure. Our problem is how to combine a representation of the global structure with a representation of the local structures.

4.1 Representation of Global Structure

We adapted Treemap[14, 15] to express the global structure of tickets. A rectangular area is assigned to a ticket or a group of tickets. Treemap can represent quantitative data by the sizes of the rectangles. When we assign the same weight to all tickets, we can understand the number of tickets in a group by seeing the size of the corresponding rectangle. When we regard a quantitative attribute as their weight, we see the size of the rectangle as the sum of the attribute values. For example, we can express a different work load (in other words, the number of update times) by the size of the rectangles.

OSS project tickets do not necessarily follow a moneylike concept. Therefore, we describe the number of update times as an example of quantitative data. However, when we treat activities in a company, we are certainly expected to express the budget scales of every activity.

4.2 Representation of Local Structures

To express the time change of attribute values of tickets, we developed two types of charts: Gantt charts and polyline charts.

(i) Gantt Chart A Gantt chart is a widely used chart to express the progress of projects. By placing values of an attribute vertically and time horizontally, the attribute value in a time interval is expressed as a horizontal bar. Our charts can occupy only a very narrow area. Therefore, we assign a different color to every attribute value to be able to read information simply from a bar without labels (see Figure 2). Moreover, we paint the background of the area with a lighter color of the current status.

(ii) Polyline Chart A polyline chart is a variation of a Gantt chart. It uses polygonal lines instead of horizontal bars. On a Gantt chart drawn on a narrow area, short bars sometimes become dots. To increase the visibility of the charts, we replaced the horizontal bars with polygonal lines. On a polyline chart, each point expresses an attribute value and a time the value was updated, and line segments connect such points (see Figure 3).

When the interval to the next update is short, a line segment with a steep gradient is drawn, and when the interval is long, a line segment with a gentle

gradient is drawn. Although it is not so appropriate to have change of the attribute value on a nominal scale connected by a polygonal line, in consideration of the visibility of changes, we adopted this representation on the assumption that they are drawn on narrow areas.

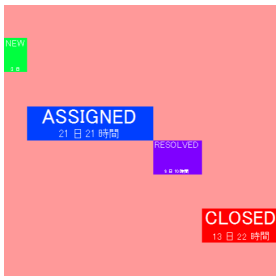

Fig. 2. Gantt chart.

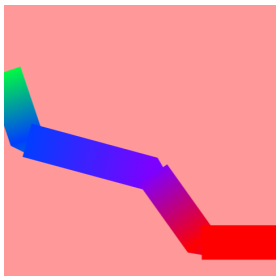

Fig. 3. Polyline chart.

4.3 Representation of a Group of Tickets

We developed three types of modes for groups of tickets: tiling, overlapping, and stacking.

(i) Tiling mode In the tiling mode, a rectangle is assigned to each ticket. A ticket chart is drawn in the rectangular area. The background color of each small chart contributes in this mode. When many tickets are displayed simultaneously, a ticket can occupy only a narrow area. Even in such a case, the observer can roughly grasp the overall situation first by observing the distribution of background colors.

(ii) Overlapping Mode We designed an overlapping mode for a polyline chart. In this mode, all charts for tickets in a group are drawn in piles in an area assigned to the group. All charts for tickets share the same time axis. We can easily grasp the tendency of time change of tickets in a group. When many polygonal lines are drawn in piles, visual confusion becomes possible. To cope with this problem, we give reverse gradation to line segments. We can read where a line segment comes from and where it goes simply by looking near both end points of the line segment.

(iii) Stacking mode By using the overlapping mode, we can grasp the density distribution of the attribute value in a certain time to some extent. However, spatial size is more suitable than the density of line segments to express how many tickets exist in a certain time. In the stacking mode, tickets with the same attribute value are collectively drawn like ThemeRiver[16]. We can get an idea of the number of tickets with each attribute value in a certain time by looking at the vertical length of the stacked bands.

Fig. 4. Tickets categorized by Product, drawn as polyline charts in the overlapping mode. The time axis is relative.

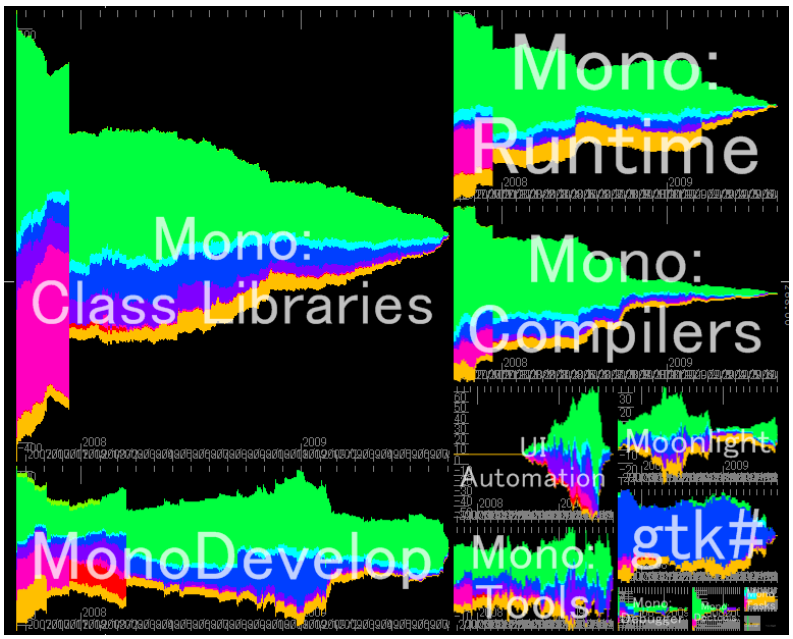

Fig. 5. Tickets categorized by Product, drawn in the stacking mode. The time axis is absolute.

4.4 Time Axis of a Group of Tickets

Whereas each ticket occupies an area in the tiling mode, two or more tickets share the same area in the overlapping and stacking modes. We prepared two types of time axis modes: an absolute time mode and a relative time mode.

(i) Absolute Time Mode The horizontal axis expresses absolute time. We can get an idea of the situation of the project during a certain period and at a specific time. For example, we can read what type of ticket existed on October 1, 2011, and how the tickets changed for three months from October 1 to December 31.

(ii) Relative Time Mode The horizontal axis expresses a relative time beginning with the start time of each ticket. We can follow how the status of tickets changed with elapsed time from the start. If the change of status has a particular pattern, it is expected that the pattern will actually be visible as well. Examples of such patterns are cases in which most tickets in a category were completed in a week or tickets in another category were neglected for one month or more.

5 Series at a Glance

We developed a tool named "Series at a Grance (SaaG)" to manipulate the visual representation explained in the previous section (see Figure 1). This section explains functions offered by SaaG.

Setup of the Global Structure SaaG shows tickets based on their hierarchical structures. For that, it is necessary to determine a hierarchical structure of tickets. A hierarchical structure can be constructed by repeating categorization based on attributes. However, since the attributes that can be used for the categorization varied according to projects, we cannot determine them beforehand. SaaG constitutes a menu of attributes used for the categorization according to ticket data. Users can specify the attribute of the first layer, the attribute of the second layer, and the attribute of the third layer with a pull down menu, respectively.

Representation of Local Structures By specifying an attribute to be expressed visually, time change of the value of the attribute is displayed in the cell of Treemap. Expression of one ticket and expression of a ticket group can be changed at any time by choosing from a menu. The option of the time-axis in the overlapping mode or the stacking mode can also be changed at any time with a menu. In the overlapping mode and the stacking mode, a group of tickets that shares an area is chosen depending on the global structure. At the stage of construction of a hierarchical structure, if only attribute for the first layer was specified, tickets would be collected in the groups of the first layer. If two attributes for the first and second layers were specified, tickets would be collected in the groups of the second layer.

Zooming Zooming function expands a specified area in the Treemap. By using this function, users can pay their attention to a group of tickets collected into a certain area on Treemap. When a user chooses an area to pay one's attention to in the rubber band by mouse dragging, the selected area is expanded to the limit of a display window. The user can repeat this operation any number of steps. One ticket can also be displayed to the limit of a display window. Cancel of the zoom operation is easy. Single click cancels last zoom operation. By this, users can easily pay their temporal attention to a part of the visualization.

Filtering Filtering function hides or shows only tickets that match some condition on attributes. By using this function, users can get some selective display, for example, users can hide expired tickets, display only tickets assigned to a certain person, and so on. Attribute values used for the conditions of filtering can be chosen from a drop-down menu. Or by clicking a specific ticket or a ticket group, all attribute values of the ticket or all common attribute values of tickets in the group are used as conditions of filtering.

Detailed View of Each Ticket When each ticket is displayed in a separated area in the tiling mode, detailed view of ticket can be shown in a new window by double-clicking on the area for the ticket. The users can update values of the ticket through the window.

Recording Logs SaaG records all user operations. Users may add comments to the operation logs and get screenshots with the comments after a series of operations. Exploratory analysis is accompanied by trial and error. Even if an analyst found some useful knowledge, it would be difficult to remember the process to the knowledge. Recording all operations with annotated comments makes analytical tasks more efficient and valuable. Furthermore, screenshots with comments help to review the processes.

6 Case Study

To verify the feasibility of the representation technique and SaaG, we performed a ticket analysis. The objective of the analysis is the Mono project [3]. The analyst is a person outside the project who explores features of the project.

The Mono project is an open-source software (OSS) project in which software for realizing an environment compatible with .Net Framework is developed according to the Ecma standard. The project is divided into several subteams for products.

The analyst observed about 20,000 tickets in search of getting to know the background of the project, particularly by seeing what type of subteams comprise the project. The analyst selected the attribute Product as the first layer of the hierarchical structure and then drew a polyline chart in overlapping mode (see Figure 4).

[3] http://www.mono-project.com/Main_Page

In the Class Libraries group, there are many line segments with steep gradients from green to purple. These segments express tickets changed from *New* (green) to *Resolved* (purple) in a short period. There are also many line segments going to and from yellow at the bottom. These segments express tickets becoming *Needinfo* (yellow) for a while. That is, although many tickets were processed quickly, many other tickets stopped processing for a while owing to lack of information. In the Mono Develop and Runtime groups, we can see a similar tendency.

Groups with smaller areas on the screen, such as UI Automation, differ in form from the others. In a group with a small area, most segments are not far stretched toward the right. That is, it turned out that work periods were generally short. Moreover, although there are many blue segments and red segments, there are relatively few purple segments. We can interpret this to mean that many tickets became *Closed* (red) instead of *Resolved* (purple), and many of them also became *Reopened* (light blue). From these, the analyst guessed that the pace of work is generally quick, but resolved judgments were rarely received and there is much rework involved.

The analyst thought that this was a tendency found in new products, and then selected the absolute time axis in the stacking mode. It became apparent that UI Automation was a new product.

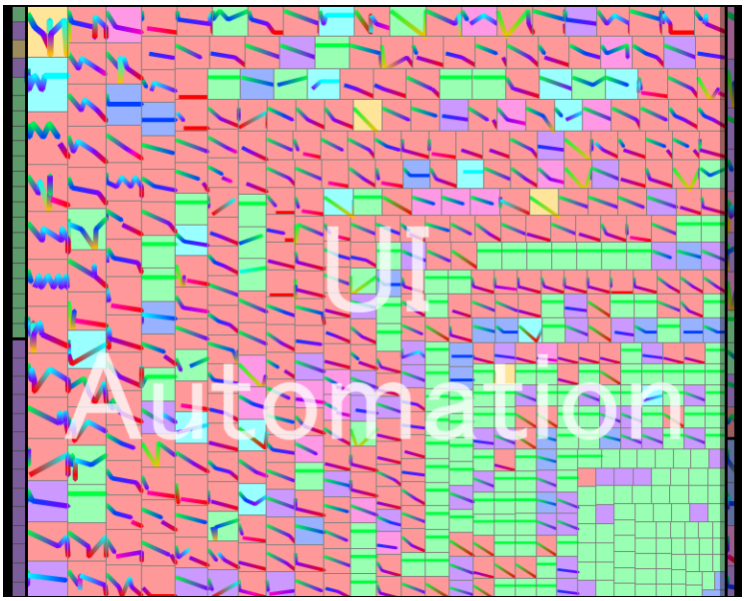

Fig. 6. Zoomed-in view of the UI Automation group. Tickets are drawn as polyline charts in the tiling mode.

To investigate UI Automation in detail, the analyst expanded the group with the zoom function and selected a polyline chart in the tiling mode (see Figure 6). The current status of most tickets is *Closed* (red) and many tickets have downward-sloping, favorable patterns. From these observations, although there are a few *Reopened* tickets, the analyst guessed that progress in this product is generally favorable. However, a few tickets exhibit a pattern of vertical vibration. These ticket tasks did not progress smoothly.

The following information about the Mono project was acquired:

1. Many Class Library tasks were solved in a short time.
2. Many Class Library tasks failed owing to a dearth of information.
3. Many UI Automation tasks were completed in a comparatively short time.
4. For a small-scale product, the resolved judgment was not necessarily received and there was much rework involved.

7 Conclusions

We developed a representation technique for visualizing numerous activities. The developed technique embeds charts showing activities in rectangles of nodes on the basis of Treemap. Both quantitative and temporal aspects of activities can be simultaneously seen with this representation technique. We also developed an analytical tool, SaaG, which can manipulate the representation. To our knowledge, there are no tools based on the combination of treemap views and project charts like Gantt chart. The tool targets tickets as activity data. Analysts are allowed to construct and modify the global structure of tickets by specifying their attributes. They can easily observe activity data from various viewpoints. To show simultaneously tens of thousands of tickets, each ticket is allowed to occupy only a small area. We designed representations of local structures to cope with the problem. In the tiling mode, their background colors show most important values even in a few pixels' area. In the overlapping and stacking mode, tickets in a group share an area and unveil major trends of the local structures. With the tool, a visual analysis can be performed more flexibly for tens of thousands of tickets.

Acknowledgement This work was partially supported by JSPS KAKENHI Grant Number 22500081.

References

1. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of IEEE Symposium on Visual Languages, pp. 336–343 (1996)
2. Storey, M.A.D., Čubranić, D., German, D.M.: On the use of visualization to support awareness of human activities in software development: a survey and a framework. In: Proceedings of the 2005 ACM symposium on Software visualization (SoftVis '05), pp. 193–202, ACM, New York (2005)

3. Ball, T., Eick, S.G.: Software visualization in the large. Computer 29, 4, 33–43 (1996)
4. Froehlich, J., Dourish, P.: Unifying artifacts and activities in a visual tool for distributed software development teams. In: Proceedings. 26th International Conference on Software Engineering (ICSE 2004), pp. 387–396 (2004)
5. Fischer, M., Gall, H.: MDS-views: Visualizing problem report data of large scale software using multidimensional scaling. In: Proceedings of the international workshop on Evolution of large-scale industrial software applications (ELISA). pp. 110–121 (2003)
6. Ellis, J.B., Wahid, S., Danis, C., Kellogg, W.A.: Task and social visualization in software development: evaluation of a prototype. In: Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07), pp. 577–586, ACM, New York (2007)
7. Ogawa, M., Ma, K.L.: Software evolution storylines. In: Proceedings of the 5th international symposium on Software visualization (SOFTVIS '10), pp. 35–42, ACM, New York (2010)
8. Ogawa, M., Ma, K.L.: code_swarm: A design study in organic software visualization. IEEE Transactions on Visualization and Computer Graphics, 15, 6, 1097–1104 (2009)
9. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visualizing time-oriented data — a systematic view. Computer & Graphics, 31, 3, 401–409 (2007)
10. Reijner, H.: The development of the horizon graph. In: Electronic Proceedings of the VisWeek Workshop From Theory to Practice: Design, Vison and Visualization (2008)
11. Heer, J., Kong, N., Agrawala, M.: Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In: Proceedings of the 27th international conference on Human factors in computing systems (CHI '09), pp. 1303–1312, ACM, New York (2009)
12. Krstajić, M., Bertini, E., Keim, D.A.: Cloudlines: Compact display of event episodes in multiple time-series. IEEE Transactions on Visualization and Computer Graphics, 17, 12, 2432–2439 (2011)
13. Wattenberg, M., Viégas, F.B., Hollenbach, K.: Visualizing activity on wikipedia with chromograms. In: Proceedings of INTERACT. pp. 272–287 (2007)
14. Shneiderman, B.: Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph, 11, 92–99 (1992)
15. Bruls, M., Huizing, K., van Wijk, J.: Squarified treemaps. In: Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, pp. 33–42 (2000)
16. Havre, S., Hetzler, E., Whitney, P., Nowell, L.: Themeriver: visualizing thematic changes in large document collections. IEEE Transactions on Visualization and Computer Graphics, 8, 1, 9–20 (2002)