

## Using the magnet metaphor for multivariate visualization in Software management

Amaia Aguirregoitia  
University of the Basque Country  
La Casilla 3, 48012  
Bilbao –Vizcaya(Spain)  
[amaia.aguirregoitia@ehu.es](mailto:amaia.aguirregoitia@ehu.es)

J. Javier Dolado  
University of the Basque Country  
Pº Manuel de Lardizabal 1, 20018  
Donostia(Spain)  
[javier.dolado@ehu.es](mailto:javier.dolado@ehu.es)

Concepción Presedo  
University of the Basque Country  
La Casilla 3, 48012  
Bilbao –Vizcaya(Spain)  
[conchi.presedo@ehu.es](mailto:conchi.presedo@ehu.es)

### Abstract

This paper presents SoftMagnet, a new multivariate analysis model for controlling and managing the processes of software project development. SoftMagnet uses metaphors and visual representation techniques to explore several key indicators in order to support problem detection and resolution. The resulting visualization addresses diverse management tasks, such as tracking of deviations from the plan, analysis of patterns of failure detection and correction, overall assessment of change management policies, and estimation of product quality. The proposed visualization uses a metaphor with magnets along with various interactive techniques to represent information concerning the software development process and to deal efficiently with multivariate visual queries. This paper shows the final implementation of SoftMagnet in JavaFX with data of a real project as well as the results of testing the tool with the aforementioned data.

**Keywords:** Information Visualization, Data visualization Software, Software visualization, Visual knowledge discovery.

### 1. Introduction

Managing software development processes is a very difficult task. Often projects are behind schedule and the resulting applications lack quality. Managers require different types of data, for instance written reports from project managers or software metrics like productivity, quality, adherence to schedule and budget. To assist in

exploration and analysis of the high volumes of data required, our research focuses on the design of a tool to efficiently process visual queries on the key measures for software development management.

Most of the data from software engineering science are multivariate, containing more than three attributes. Therefore, multivariate information visualization techniques should be considered. The use of multivariate information visualization techniques is intrinsically difficult because the multidimensional nature of data cannot be effectively presented and understood on real-world displays, which have limited dimensionalities [1]. However, these techniques are very useful and offer powerful insights into the decision-making process.

The present paper describes SoftMagnet, which uses a metaphor with magnets as well as various interactive techniques, and applies the benefits of multivariate information visualization techniques to the Software management field. The magnet metaphor is intuitive, which facilitates learning and interacting with our multivariate information visualization.

The next section presents the related work. Then a description of the tool is given. In the conclusion, we assess our work and present the major benefits of the proposed visualization.

### 2. Related work

Much work has been conducted on visually presenting multidimensional data with the same underlying goal: to display complex, multidimensional information using a lower (e.g., two- or three-) dimensional space in a form

suitable for understanding [2]. There are a large number of techniques but we will now mention just the few related to our work.

One of the more popular multivariate visualization techniques for Statistics is the scatterplot matrix which presents multiple adjacent scatterplots. Scatterplots are bivariate or trivariate plots of variables against each other and the power of the scatterplot matrix lies in its ability to show the internal connections of the scatter diagrams [3].

The idea of pairwise adjacencies of variables is also a basis for the Hyperbox [4] where all possible pairs of variables are plotted against each other and any pair can be brought to front with Cartesian axes with all others still visible. Some other related visualizations are hierarchical axis [5] [6] and HyperSlice [7].

Parallel coordinates is another extended technique, which uses parallel axes instead of perpendicular to represent dimensions of a multidimensional data set [8, 9]. Star plots, star coordinates, polar charts, height maps, stacked displays, heat maps, table lens and time wheels are more recent but quite widespread techniques.

In the software area multivariate visualization has been used to evaluate and improve synthetic tests, compare test suites and assess bug reports [10], to study software evolution and the complex correlations of its attributes [11], to analyze execution information and exceptions [12] or to provide integrated condensed graphical views on source code and release history data using Kiviat diagrams [13].

The approach in this paper was inspired by Dust and Magnet visualization [1] and DataMeadows [14]. It uses a metaphor with magnets and various interactive techniques to represent software development process information and to deal with multivariate visual queries efficiently.

### 3. Softmagnet project

#### 3.1. The use of metaphors

Metaphors are important tools in information visualization as they provide familiar cognitive models to help users to browse unfamiliar information spaces [15]. Six advantages of Visual Metaphors have been described in previous works: (1) to motivate people, (2) to present new

perspectives, (3) to increase remembrance, (4) to support the process of learning, (5) to focus attention and support concentration of the viewer, (6) to structure and coordinate communication [16].

A familiar visual metaphor can lower the cognitive load imposed on a user and increase the rate of comprehension. The Dust & Magnets technique is an illustration of the use of metaphors to implement a visualization with a high power of interaction for exploration, and it exemplifies how a simple interaction can provide important insights into a complex data set through animation [1].

#### 3.2. The measures for project management

Previous field studies and a questionnaire conducted with several Basque companies produced the framework for metric definition and selection. We used the most relevant aspects found in these surveys to guide the definition of the measures in the system. These surveys highlighted the importance of scheduling and project estimation tracking, the importance of management of requirements changes, the relevance of risk identification and analysis, and the benefits of failure identification, classification and correction.

The list of measures considered in the proposal (for each task involved in the development process) is presented in *Table 1*.

#### 3.3. Description of the tool

The visualization has been implemented using SDK JavaFX Preview Release 1; and the general layout of how Softmagnet applies the metaphor is presented in *Figure 1*. It shows an overview with real data of an EIS (Executive information System) development project with 80 tasks.

The main window presents the display area and a control panel on the right side, which can be rendered invisible using the menu. The panel is used to set up the visualization and the display area shows the information of the selected measures using the magnet metaphor.

Using the control panel the user selects the number of graphs to be displayed simultaneously.

Table 1. List of measures

Task effort	Number of Documentation reviews
Estimated task effort	Number of other type reviews
Task cost	Number of total reviews
Planned task cost	Requirement failure correction effort
Number of Requirement failures	Design failure correction effort
Number of Design failures	Code failure correction effort
Number of Code failures	Documentation failure correction effort
Number of Documentation failures	Other type failure correction effort
Number of other type failures	Failure correction total effort
Number of total failures	Number of changes required
Number of failures detected by the client	Number of changes rejected
Number of failures detected by the developers	Number of changes implemented
Requirement failure detection effort	Number of changes pending
Design failure detection effort	Number of deliverables planned
Code failure detection effort	Number of deliverables rejected by the client
Documentation failure detection effort	Number of deliverables accepted by the client
Other type failure detection effort	Number of pending deliverables
Failure detection total effort	Number of detected risks ( with description and type)
Number of Requirement reviews	Effort deviation
Number of Design reviews	Cost deviation
Number of Code reviews	Risk detection effort

The user can define up to four graphics and then hide and show them as required by changing the selection of “Number of graphs”. A button is activated for each graph to define the measure to be presented in that graph and the color to be used for that measure as shown in *Figure 2*.

After clicking on the corresponding button, any of the measures in *Table 1* can be selected for visualization in each graph. The name of the measure selected for each graph is displayed in the right lower part with a grey label.

The tool divides the visualization area into as many squares as graphs have been defined. In each square the tasks are arranged initially in a diagonal line depending on the value of the selected measure for that graph. As a data point (task) is located along the diagonal according to its value, a data point can be located closer to or farther from a magnet according to its value.

*Figure 1* shows four squares with four different graphs where the user can analyze the four selected measures simultaneously (effort deviation, total failure correction effort, number of

changes implemented and total number of errors). When the task with the highest deviation (Second phase coding) is selected in graph 1 the tool highlights that that same task is the one with the highest number of errors and failure correction effort, which seems to indicate that the deviation may be caused by failures and failure correction in the codification phase. By looking at the numbers in the axes the user finds that 87 hours have been dedicated to error correction and that the deviation of the task is 136. To further analyze the failures the user can choose a concrete type of error, as “Code errors” instead of “Total number of errors” from the indicator list and visually compare the new results.

*Figure 3* shows the same configuration of graphs and settings but in this case, another phase (Coding of International Module) is selected. When the user selects this task, the user can see in graph 2 (and using the data table if required) that only 28 hours have been dedicated to error correction when the effort deviation for the task is 92. The tool highlights in graph 3 (lower right) that that task has the highest number of modifications implemented. This information

helps the user to detect a possible cause of the deviation.

As the magnet is being dragged, the “particles” (tasks) are attracted and move with the magnet. The user can use this feature to easily compare the values for different measures by setting magnets in a way that tasks are arranged in parallel lines. Scales and axes are set automatically by the visualization and they are independent for each square. There is highly valuable global information contained in the positions of the tasks. *Figure 4* shows two graphs where both magnets has been dragged from the upper left corner to the upper left corner so that the tasks are presented vertically.

An interesting feature is that when the user selects one of the tasks, the visualization searches for that task in the rest of the graphs. It indicates its position in all the graphs with a colored dot and

the user can visually analyze the data of the different measures simultaneously (See *Figure 4*).

One of the disadvantages of the representation is that more than one task can have the same value for a measure, which results in a graph where a dot represents a set of tasks instead of a unique task. This situation is indicated in the graph with the label “SET” instead of a task description in the dot (See *Figure 4*). However, the identification and descriptions of the selected tasks are always visible in the square in the lower right part of the screen as shown in *Figure 5*.

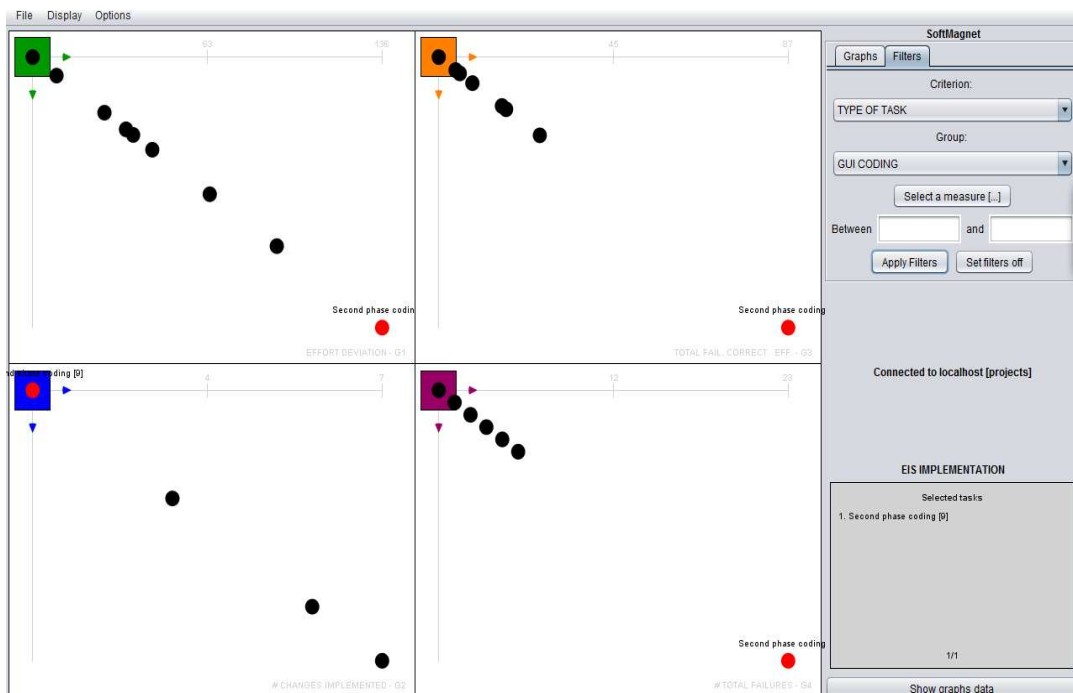


Figure 1. Overview of the system.

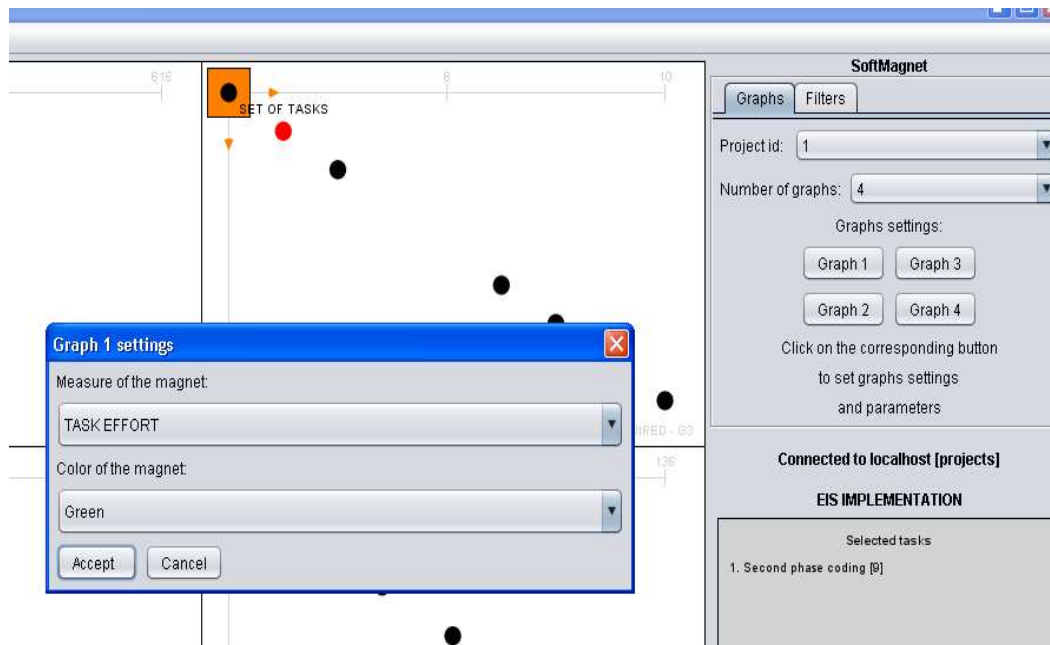


Figure 2. Graph definition after clicking on Graph1

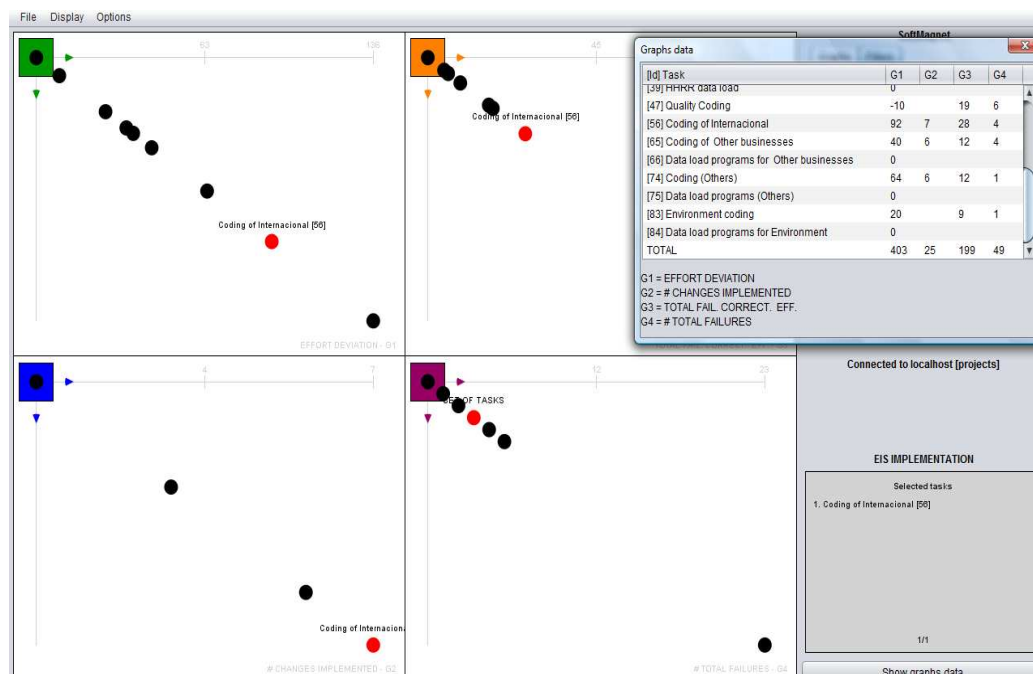


Figure 3. Data analysis by selection.



Figure 4. Dragging the magnets to position the tasks

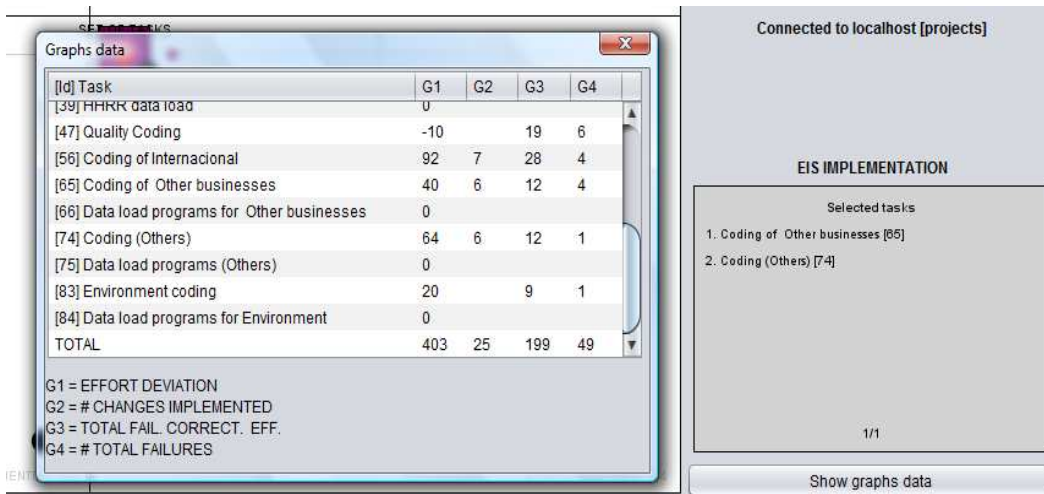


Figure 5. Descriptions of the selected tasks and data for the visible tasks and measures.

The visualization offers the information for all the tasks and all the measures in the actual representation as text in a data table. Figure 5 presents the table that can be accessed by clicking on “Show graphs data”. The data table incorporates a final row with the total for each column (measure) and the actual filters also

applied to the data table. The table presents the task identification, description, and four columns with the data of the four actual measures.

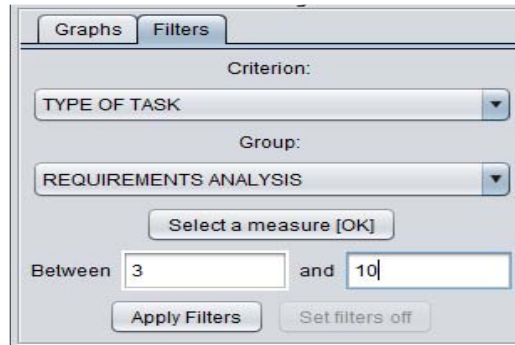


Figure 6. Filtering options

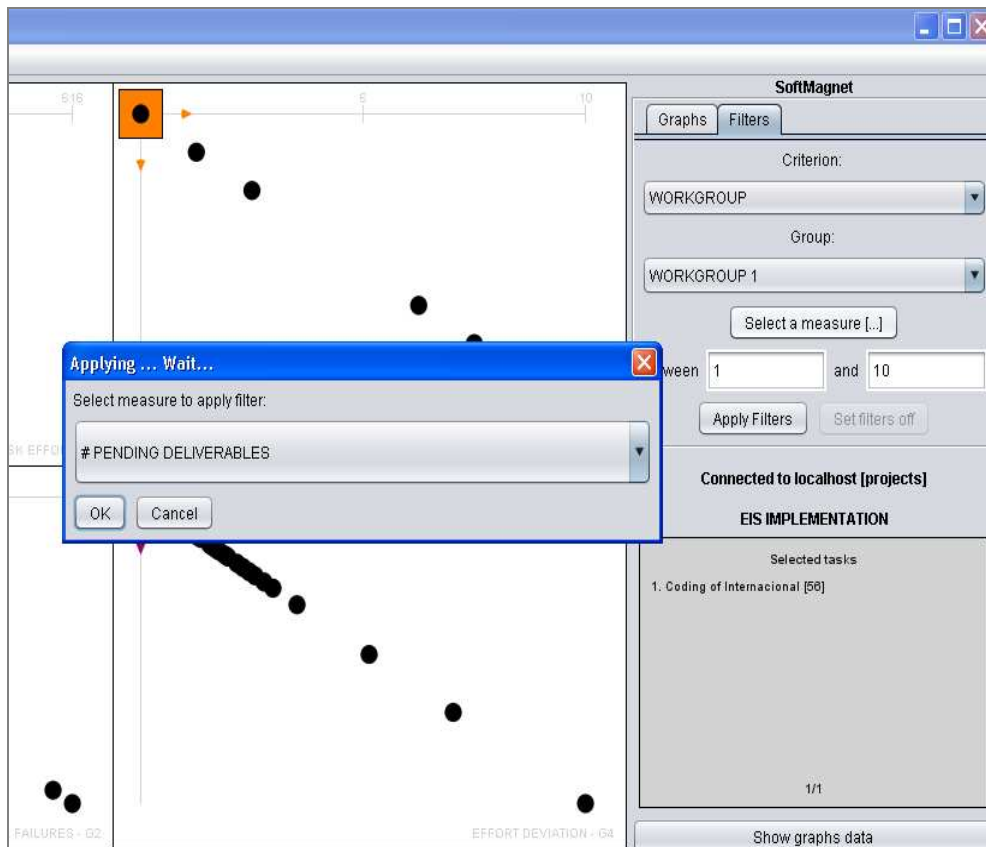


Figure 7. Applying a filter by value

In order to find out more about a certain group of tasks the system implements filters using different hierarchies or criteria or the filter by value option. Tasks of the project are classified according to different hierarchies (type of task, development module, workgroup and project phase) and for each hierarchy there are different groups. For example “*Requirement analysis*” is a group or level of the “*Type of task*” hierarchy. The tool includes a feature in the “*Filters*” tab to apply a filter based on the stated groups.

Another interesting feature is the “*Filter by value*” option which allows specifying a measure and a range of values to filter and to display the set of tasks meeting that condition as shown in *Figure 7* (in the figure only the tasks assigned to workgroup one with pending deliverables will be displayed). It is very easy to filter, for example, only those requirement analysis tasks with modifications and then analyze the measures in the graphs: effort deviation, modifications rejected, pending modifications and number of total failures. The user can “set filters off” at any time or apply multiple filters simultaneously. The axes for each graph are recalculated and redrawn after applying a filter for the best possible visibility.

#### 3.4. Benefits of the visualization

Each graph of the visualization allows for exploration of how the different tasks contribute to the total value of a measure and its distribution. This can be useful, for instance, for error distribution analysis and error pattern detection. The visualization can present up to four measures simultaneously. Therefore, if the user needs to explore different variables simultaneously it is possible to compare values and distribution of related measures in the same visualization. An example, if the user selects four graphs and different types of failure as the selected measure for each graph, it is possible to analyze error distribution by type as well as the distribution and pattern for each error type. Multiple-variable visualization can be useful for finding relationships between variables. As an example, the user can examine the measures “Failure detection effort” and “Number of errors detected” and analyze the relationships between these measures.

The tool allows the user to identify problematic areas. When measures such as “number of errors”, or “Effort deviation” are selected, the user can detect very easily which tasks are problematic. With a quick look at the right or lower (depending on the position of the magnet) portion of the graphs, the user can identify which tasks have the highest values of these metrics.

Another benefit is that when a task stands out from the others on the graph of one measure, the tool searches and highlights the value of that same task for a different measure on a separate graph. As seen in the aforementioned example, if effort deviation is presented in one square and number of changes implemented in another square, when the user selects, let us say, the task with the highest effort deviation in the first graph, the other graph will highlight the position of that task for the “numbers of changes implemented” variable. The user can easily detect if the high values of effort deviation correspond to high values of approved modifications to the software and analyze if those effort deviations could have been caused by those modifications.

Furthermore, the user can move from an overall view to a detailed view of a measure by clicking on one of the values. When a circle in one of the graphs represents a set of tasks with the same value the user can access a detailed list of tasks by selecting the circle.

In addition, the user can focus on a certain set of tasks using the specified classifications. All the tasks are classified according to different criteria which are used to filter the information and facilitate analysis. The established classification to analyze the tasks of our project is as shown in *Figure 8*.



Figure 8. Task classification for an EIS development

The user can employ these classifications to apply filters and perform actions such as: analyzing the errors of only the set of tasks performed by a certain workgroup, visualizing the status of the deliveries of only one of the modules or evaluating the cost deviation of a certain type of task, like programming or documentation, to assess if there have been estimation problems with that type of task.

Additionally, the filter by value option lets the user specify a measure and a range of values to limit the information presented. With the tool, the user can apply a filter based on the value of a certain measure and graphically explore up to four measures different from the one used for the filter. The graphs will only display the tasks with the value specified by the filter. This can be useful, for example, when the user tries to focus on the tasks that have more than two deliverables pending, or on those that have received more than three modifications or on those that have been associated with a risk.

Moreover, by selecting the “show data” option, the user can examine the values for each of the tasks and see the total for the visualized data in text format. If the user has

previously applied a filter, it applies not only to the graphs, but also to this total.

Table 2 summarizes the benefits of the visualization.

Table 2. Benefits of the visualization

<p>The representation offers multiple possibilities of arranging clusters of tasks and allows for visualization of certain chosen subsets of tasks.</p>
<p>It integrates a high interaction level (multiple selections, mouse functions, pop-up labels, ..) and includes features such as a general view, additional information for a task on mouse click (the task is highlighted in the other graphs) or magnet drag facilities.</p>
<p>Softmagnet presents up to four measures in a single view. The capability of showing multiple measures at a time is a central feature because it lets the user analyze, detect patterns and draw a conclusion from information of multiple variables. Softmagnet is adept when the analysis includes different measures simultaneously.</p>
<p>It includes a complete set of measures for analysis and offers the possibility of filtering of the information according to the value of a selected measure (filter by value). This feature, along with the four graphs, allows consideration of five measures at the same time.</p>
<p>The visualization presents visually, as well as and in a data table, the numerical value of more than one measure for multiple tasks in the current visualization. It is possible to study all the tasks in the visualization simultaneously with the numerical information for more than one measure in view.</p>
<p>The approach calculates and immediately presents the totals for all the visible tasks and actual measures.</p>
<p>The proposal includes a “Filter by group” option for visualizing a subset of tasks, which allows visualizing only the selected groups of tasks simultaneously.</p>
<p>Softmagnet has drag options to position the magnets and tasks, which is valuable when comparison between different measures is required.</p>
<p>The axes are scaled according to the presented values. Therefore, when the user wants to focus on a set of values (similar to a zoom on an area) he can apply a filter to that measure and those values and the graph will be automatically redrawn.</p>
<p>Softmagnet includes graphical and textual information to complement the graph.</p>
<p>It concentrates on detecting problematic areas by focusing on nonstandard or irregular values.</p>

#### 4. Conclusion

This paper presents a representation for multivariate visualization and reasoning about datasets from the software project management area. It combines filtering and selection options and visualizes multiple measures for comparison. The different hierarchies and levels, the drag options, the task selection and search capabilities, and the wide range of possibilities for analysis are the most salient strengths of the project. In our future work, we will perform user tests to further explore the visualization and to improve it.

#### Referencias

- [1] J. Yi, R. Melton, J. Stasko *et al.*, "Dust & Magnet: multivariate information visualization using a magnet metaphor," *Information Visualization*, vol. 4, no. 4, pp. 239-256, 2005.
- [2] P. Wong, and R. Bergeron, "30 years of multidimensional multivariate visualization," *Scientific Visualization, Overviews, Methodologies & Techniques*, pp. 3-33: IEEE, 1997.
- [3] W. Härdle, and L. Simar, *Applied multivariate statistical analysis*: Springer, 2007.
- [4] B. Alpern, L. Carter, I. Center *et al.*, "The hyperbox." pp. 133-139.
- [5] T. Mihalisin, J. Timlin, and J. Schwegler, "Visualization and analysis of multi-variate data: a technique for all fields," in Proceedings of the 2nd conference on Visualization '91, San Diego, Ca, USA, 1991, pp. 171-178.
- [6] T. Mihalisin, J. Timlin, J. Schwegler *et al.*, "Visualizing multivariate functions, data, and distributions," *IEEE Computer Graphics and Applications*, vol. 11, no. 3, pp. 28-35, 1991.
- [7] J. van Wijk, and R. van Liere, "HyperSlice: visualization of scalar functions of many variables," in Proceedings of the 4th conference on Visualization'93, 1993, pp. 119-125.
- [8] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. 1, no. 4, pp. 69-91, 1985.
- [9] A. Inselberg, B. Dimsdale, I. Center *et al.*, "Parallel coordinates: a tool for visualizing multi-dimensional geometry." pp. 361-378.
- [10] D. Leon, A. Podgurski, and L. White, "Multivariate visualization in observation-based testing." pp. 116-125.
- [11] L. Voinea, and A. Telea, "Multiscale and multivariate visualizations of software evolution," in Proceedings of the 2006 ACM symposium on Software visualization, 2006, pp. 115-124.
- [12] A. Orso, J. Jones, and M. Harrold, "Visualization of program-execution data for deployed software." pp. 67-76.
- [13] M. Pinzger, H. Gall, M. Fischer *et al.*, "Visualizing multiple evolution metrics," in Proceedings of the 2005 ACM symposium on Software visualization, 2005, pp. 67-75.
- [14] N. Elmquist, J. Stasko, and P. Tsigas, "DataMeadow: a visual canvas for analysis of large-scale multivariate data," *Information Visualization*, vol. 7, no. 1, pp. 18-33, 2008.
- [15] B. Shneiderman, S. Card, and J. Mackinlay, *Readings in Information Visualization: Using Vision to Think*: Morgan Kaufmann, 1999.
- [16] M. Eppler, "The Image of Insight: The Use of Visual Metaphors in the Communication of Knowledge," in Proceedings of I-KNOW '03, Graz, Austria, 2003, pp. 81-88.