

# AN ENHANCED INTERACTION AND FILTERING WIDGET

Yasmin Al-Zokari<sup>1</sup>, Peter Dannenmann<sup>2</sup>, Hans Hagen<sup>3</sup>, Dirk Zeckzer<sup>1</sup>

<sup>1</sup>*Leipzig University, Augustusplatz 10, 04109 Leipzig*

<sup>2</sup>*RheinMain University of Applied Science, Rüsselsheim, Germany*

<sup>3</sup>*TU Kaiserslautern, Erwin-Schrödinger Straße, Kaiserslautern*

## ABSTRACT

In information visualization, interaction mechanisms are an essential factor to the success of visualization. They should be easy to use and useful for filtering, exploring, and analyzing the data visualized. Usually, direct manipulation on the visualizations is the most important aspect that researchers consider. However, users still tend to use controls (widgets) for some interaction, especially, when manipulating (e.g., filtering) the data. During our work with users on the visual representation of safety analysis results, we found limitations in the interaction capabilities of sliders. In order to overcome these limitations, we introduce our Range-Selection-and-Filtering RSF slider, which enables users to work more efficiently when selecting relevant data from large data sets compared to conventional slider designs by enhancing the ease of use and increasing its usefulness. Additionally, we applied this slider to different application domains and evaluated it.

## KEYWORDS

Graphical user interfaces, Interaction styles, Interaction techniques, User interfaces.

## 1. INTRODUCTION

In the “Safety Assessment and Risk Analysis” domain, there is a strong need to efficiently visualize analysis results. A safety-critical system could cause different levels of harm to users --i.e., injury, death, etc.-- or to the environment (Savive Pty Ltd, n.d.). The data is usually divided into several categories, e.g., unacceptable (high), tolerable (moderate), and negligible (low) risks (Bozzano & Villafiorita, 2010, p. 68). Safety analysis techniques and procedures are used to detect risks and hazards for removing, avoiding, or reducing the probability of the failure of a system (Bozzano & Villafiorita, 2010, p. xiii). Visualization enables engineers to efficiently analyze the resulting data. Usually, direct manipulations on the visualizations are the most important aspect that researchers consider. Filtering is required by the tasks of safety analysts, because they search for the most safety critical events and prioritize them. Therefore, in this paper, we investigated and illustrated the strengths and weaknesses of different types of sliders and propose the Range-Selection-and-Filtering RSF slider for Minimal Cut Set (MCS) analysis. Sliders are an easy means to filter data and provide a sensitive mechanism for specifying values (Eick, 1994, pp. 119–120). Therefore, they are very well suited for supporting the analysts in their daily work.

Section 2 describes the requirements; Section 3 describes the state-of-the-art sliders. In Section 4, we describe our RSF slider. Section 5 provides the ease of use assessment. Section 6 shows two case studies for assessing usefulness and usability of our slider. Finally, Section 7 summarizes our findings and presents the future work.

## 2. REQUIREMENTS

In this paper, we present, describe, and discuss our RSF slider widget for interacting with numerical data. The assumptions and requirements are:

- (a) Manipulation of multiple ranges: We assume that the numerical data should be categorized using  $n$  adjacent intervals

$$I = \{I_i | i = 1 \dots n\} \quad (2.1)$$

$$I_i = [l_i; r_i]; \quad i = 1 \dots n-1 \quad (2.2)$$

$$I_n = [l_n; r_n] \quad (2.3)$$

$$l_j = r_{j-1}; \quad j = 2 \dots n \quad (2.4)$$

- (b) Provide the data distribution: We need to provide the distribution of the data.  
(c) Support of color schemes for the categories: A distinct color is associated with each category (interval). This color can be used in coordinated views as representation of its category.  
(d) Space efficiency: The widget should use a minimal amount of space.  
(e) Easy to use: number of interaction steps to reach the required goal.

These requirements were derived from the applications described in Section 6. The color scheme (c) can be adapted to support people having color view deficiencies or color-blind visions (Ware, 2004, p.99). The selection of ranges according to (a) implies that the uncategorized data below  $l_1$  and above  $r_n$  is not displayed. This means, the widget serves also as filtering widget. In this paper, we propose the RSF slider having the required properties (a) - (d).

### 3. RELATED WORK

Sliders were initially used by selecting a value for filtering the data (Section 3.1). Besides filtering according to single values, they can also be used to filter according to ranges (Section 3.2). The range sliders were then augmented with histograms to provide an overview over the distribution of the values (Section 3.3 and Section 3.4). In general, sliders consist of the following elements: one or more thumbs; a slider body, i.e., rectangle: the space between the thumbs; ticks, i.e., dividers: used for showing the values; and labels, i.e., text describing the values. These elements vary in number and shape. Some sliders also include additional elements.

#### 3.1. Standard or Single-Thumb Sliders

Standard sliders are widely used in many applications, e.g., for showing the history of legal cases (Harris, et al., 1999), for finding accommodations (HomeFinder) (Williamson, et al., 1992), or selecting names (Ahlberg & Shneiderman, 1994a), and are provided by GUI libraries like Java Swing or the InfoVis toolkit (Fekete, 2004). This type generally consists of one thumb, a rectangular body, several ticks, and labels for the ticks. These sliders are mainly used for filtering by selecting a value. However, they can easily be combined to select ranges like in Figure 1. The main issues with these sliders (Al-Zokari, et al., 2011c) are: they use a large amount of screen space, which violates Requirement (d); they do not support colors in an obvious way (Requirement (c)); no information about the data or data distribution is provided (Requirement (b)); they are not easy to use which violates Requirement (e); and it is not easy to interact with them (Al-Zokari, et al., 2011a), because they are multiple (four in our case) and the user has to think while choosing the correct slider for the specific required range.

#### 3.2. Range Sliders

Range slider (Ahlberg & Shneiderman, 1994b), (Tanin, et al., 1996) are also called dual-thumb, min-max, or dynamic-query sliders. Many approaches adopted this type of slider because of its powerful representation, such as Dyna-TreeMap (HCI Lab, U.S. Bureau of Census, n.d.), the dynamic queries of (Dang, et al., 2001), Ymap, Dotfire, or GRIDL (Shneiderman, et al., 2000), FilmFinder (Ahlberg & Shneiderman, 1994b), HomeFinder (Williamson, et al., 1992), Spotfire (Plaisant & Laboratory, 1993), (Ahlberg, 1996), (Card & Mackinlay, 1997), and the prefuse toolkit (Heer, et al., 2005)).

This slider has a second thumb allowing to select ranges instead of single values. However, thumbs can overlap (Figure 2). In this case, the user can only move the upper thumb. If the thumb movement is constrained, then this upper thumb can only be moved in one direction. For example, if the low thumb is over the high thumb then the user can only move towards the left, otherwise to the right. Therefore, the user cannot freely decide which way to move the thumbs and has no full control over the slider.

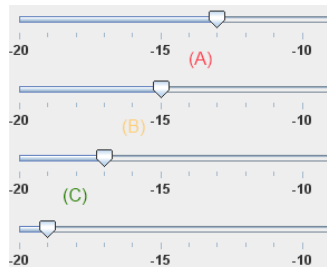


Figure 1 Four standard sliders defining the three ranges (A), (B), and (C).

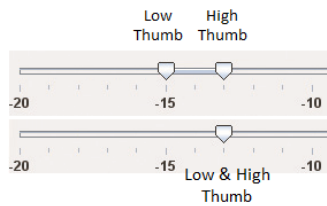


Figure 2 Thumbs overlap.

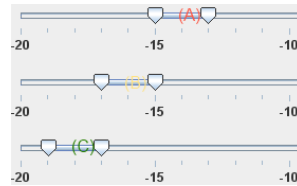


Figure 3 Three dual-thumb sliders defining three ranges (A), (B), and (C).

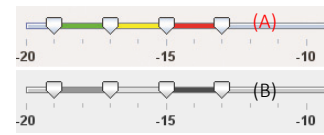


Figure 4 (A) Multi-Thumb slider with background colors. (B) As seen by users with total color blindness.

This becomes challenging for more than two thumbs. As in the case of the single-thumb slider (Section 3.1), multiple range slider can be combined to select ranges like in Figure 3, where they are used to select the three ranges marked (A), (B), and (C). This combination fulfills the requirements (a) and (c). However, the solution still uses a large amount of screen space between the sliders, violating Requirement (d). Moreover, no information about the data or data distribution is provided violating Requirement (b). Solving the violation of Requirement (d), lead us to the colored multi-thumb slider shown in Figure 4 (A). This slider is implemented using and adapting the Swing library (Swing, n.d.). However, for users with color vision deficiencies, it is difficult to distinguish between the ranges and to map them to the main visualization (Figure 4 (B)). To enhance the flexibility of the slider, we have implemented it to be dynamic according to the needs of the users by being able to change the number and the colors of range. In addition of having the fixed coloring scheme for the full color vision and the deutanope and tritanope color deficiency visions (Figure 5).

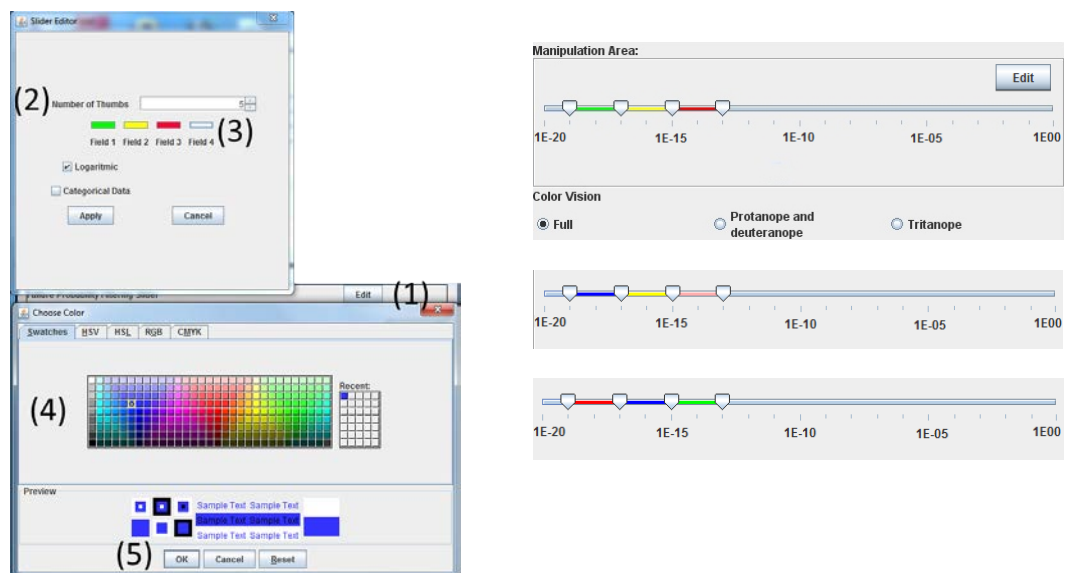


Figure 5 A) Adaptive coloring for the ranges and the number of thumbs.

B) The fixed color schemes for different color visions.

### 3.3. Interactive Legends

A special case of range sliders are the interactive legends by (Henry, et al., 2010) and (Tweedie, et al., 1996). This type provides information about the data visualized, i.e., size. However, the issues from the dual thumb slider are inherited (Section 3.2). The interactive legends with multiple handles improve the direct manipulation for interaction, especially selecting ranges. The additional improvements on the main elements of the slider were, that there are no restrictions on the number of thumbs, also called handles, and that the background is not rectangular. However, the handles still can overlap.

### 3.4. Histogram Sliders

(Li & North, 2003) added an internal histogram to the dynamic query slider and (Li, et al., 2003) compared these two approaches. It provides information about the distribution of the data being analyzed inside the slider's rectangle. The limitations of this type of sliders are: total overlapping of thumbs, the histogram is squeezed into the small slider's rectangle space, it is hard to distinguish the different levels, and the thumbs cover the histogram.

(Tweedie, et al., 1996) placed the histogram over the slider and show the relations within multi-attribute data sets. The limitations of this type of sliders are: total overlapping of thumbs, the information of the histogram outside the range is hidden when the thumbs move towards the middle (no overview information).

In general, the main elements of the histogram sliders are: two thumbs, a rectangular background (slider body), and a histogram placed inside or outside the rectangle. The histogram slider introduced by (Eick, S. G., 1994, pp. 119–120) used the rainbow color map. His slider has two more elements over the main ones: a barplot for discrete data and a density plot for continuous data. An interesting mechanism is, that the user can select/deselect arbitrary regions and get a thumb. Additionally, it shows the overview of the data being analyzed. The limitations of this type of sliders are: rainbow coloring for exploring data is “still considered to be harmful” (Borland & Taylor, 2007); the thumb is not recognizable; when the bar is gray and when it is over a light blue or orange color, it is difficult to recognize by the user (low contrast); and interaction for zooming is missing. The information is only visible when the mouse is over the bar, i.e., an overview is not available. However, if the total information was visible all the time, this slider would be very crowded. Another problem is that this type supports only one range. Therefore, it does not support the analysis with multiple ranges.

## 4. THE RANGE-SELECTION-AND-FILTERING (RSF) SLIDER

To fulfill all requirements stated in Section 2, we created the Range-Selection-and-Filtering (RSF) slider (Al-Zokari, et al., 2011b). This slider is a combination of previously available sliders (Section 3) with the following extensions: thumbs are not overlapping each other, thumbs do not overlap information.

We started with a multi-thumb slider as baseline (Figure 4). It consists of four thumbs on a rectangular body with ticks and labels. The thumbs were enumerated from 1 to 4 (left to right) and they were constrained such that if  $i < j$  then  $thumb_i$  is left to or at the same position as  $thumb_j$ . Each pair of thumbs ( $i; i+1$ ) designates a range. In the given example, there are four thumbs defining three ranges. This slider has the following advantages: more space efficient as shown in Figure 1 and Figure 3, the categories can be mapped to the background colors of the ranges (area between two thumbs), the different color schemes can be used supporting users having color deficiency problems, and it is easy to understand.

However, the following issues needed to be resolved: overlapping thumbs cannot be recognized, overlapping thumbs can only be moved in the direction of the upper thumb (due to the range constraints). Allowing to move both thumbs does not solve the problem as then we need additional modifier keys to decide how many thumbs we want to move: only the upper one, separating it from the others, two, three, or more. To avoid the overlap of thumbs, we positioned them at different distances to the main axis (Figure 6): the position of the leftmost thumb ( $W$ ) is lower than the position of thumb ( $X$ ), and so on. Thus, the values  $V$  of the thumbs are always ordered:  $V_w < V_x < V_y < V_z$ . In addition, the thumbs are elastic in the sense of thumbs moving together. If, for example, thumb  $Z$  is moved to the left then eventually it will be at the same horizontal position as thumb  $Y$  and the range  $[V_Y; V_Z]$  will become empty. Moving the thumb  $Z$  further to the left will also move thumb  $Y$

keeping it vertically below the Z thumb (Equation 1). This holds for all movements of any thumb to the left and to the right.

$$\text{Equation 1} \quad \text{Position}(Y)_{Thumb(i)} = 12 - 8 * i \quad \text{where } i \text{ is the } i\text{-th thumb}$$

Nevertheless, after applying this approach, to enhance the understandability of the thumbs position, we decided the following: the thumbs should point to the slider body, and the space between the leftmost/rightmost thumbs and the rectangle of the slider should be reduced (Requirement (d)).

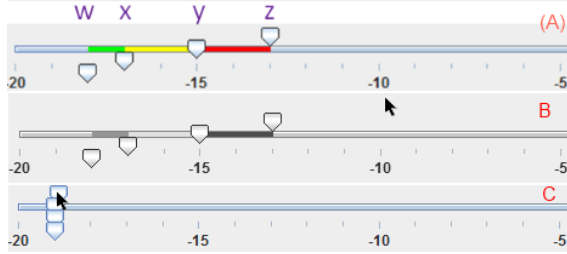


Figure 6 (A) The names of the thumbs of the RSF slider.  
(B) Clear importance even with total color blindness.  
(C) No overlapping thumbs problem.

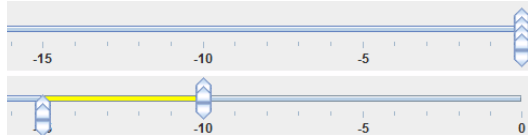


Figure 8 Double thumb slider problem (neither easy to use nor understand).



Figure 7 Slider icons. a) Java standard icon for the thumb. b) Our thumb design to point in two directions. c) Our thumb design to point in one direction with a pointer for both w and z thumbs. d) Our final thumb design.

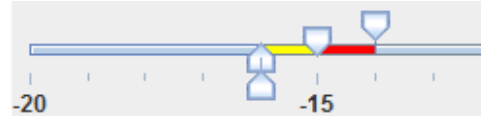


Figure 9 The problem when using the pointing thumbs



Figure 10 Final slider icon design (even number of thumbs).



Figure 11 Final slider icon design (odd number of thumbs).

Therefore, we changed the thumb icons to the ones in Figure 7d. To let the thumbs always point to the slider body, the icon of the thumbs are changed depending on the value of the thumbs. This means, when the number of thumbs is odd, the shape of the middle thumb is that of a rhombus, the thumbs set for the higher values point downwards (towards the rectangle body of the slider), and the thumbs set for the smaller values point upwards. When the number of thumbs is even, half of the thumbs point downwards and the other half point upwards depending on their set values (Figure 10 and Figure 11). Alternative icon designs are shown in Figure 7b - 7c. However, as can be seen from Figure 8 - Figure 9, the other icon designs were less suitable:

- The standard Java icon: always pointing downwards Figure 7a, in contrast to requirement 1.
- The new double thumb icon: takes a lot of space and neither easy to use nor understand (Figure 7b and Figure 8).
- The new thumb with a stick icon (pointing thumb): has a problem when overlapping (Figure 7c and Figure 9).
- Our final design: easy to understand and use with no overlapping problems (Figure 7d and Figure 10).

To solve the movement restriction of the thumbs, lower thumb should not cross higher thumbs and vice versa, and make it easy to use, the thumbs are allowed to push each other. Thus, when arriving to the same position of thumb y moving thumb z to the left leads to moving thumb y to the left, and the range  $[V_y, V_z]$  becomes empty. This has a *cascading effect*: the ability of adjusting successive thumbs (or interaction handles) in the movement direction by moving only one thumb. This means, moving thumb z further to the left will also move thumb y keeping it vertically below the z thumb, and when arriving to the same position as thumb x,

thumb  $x$  is moved together with thumb  $z$  and thumb  $y$ , etc. This holds for all movements of any thumb to the left and to the right. Additionally, we added a histogram and the number of objects per label (Figure 12). Since showing text horizontally leads to occlusions, we improved it by changing the orientation of the text to be vertical (Figure 13). Thus, this allows to get an overview over the data distribution and to support the user to choose the intervals.

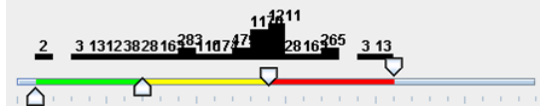


Figure 12A data set of 4158 objects with horizontal text.

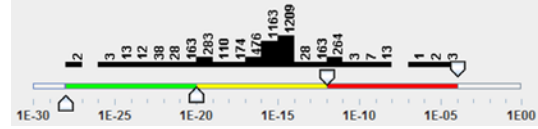


Figure 13 A data set of 4158 objects with vertical text (Final design).

Table 1 Relative Number of objects = objects in range / max{ objects in range}.

Constant Number of objects	Relative Number of objects (height)	Histogram Height
] 0, 25[	] 0, 1/5[	5
[25, 50[	[1/5, 2/5[	10
[50, 75[	[2/5, 3/5[	15
[75, 100[	[3/5, 4/5[	20
[100, 125[	[4/5, 1[	25
[125, ∞[	[1]	30

The user is immediately aware, whether data is filtered, and which part of the data is currently assigned to which category. We use histograms with fixed height and with relative height (Table 1).

The RSF slider has the following parameters:

- Number of thumbs: number of ranges
- Color scheme: individual colors of the ranges
- Use of a logarithmic scale

This allows to adapt the slider to many application needs. The advantages of the RSF slider are:

- Space-efficient visualization of the ranges using multiple thumbs (Requirement d)
- Thumbs do not overlap each other (Requirement e)
- It is easy to adjust the thumbs and thus the ranges (Requirement e)
- Usable by users with color vision deficiencies because thumbs are at logically different levels (Requirement c)
- Total overview over the distribution of the data being analyzed (Requirement b)
- Detailed information for each range and sub-range (bucket) and the data that is currently filtered (Requirements b, e)
- Allow the users to determine how much of the data is currently being analyzed (Requirement e)
- The range selection can be small (Requirement e)
- No possibility to miss data outside the filtered range (Requirement e)

The RSF slider improves the analysis capabilities by reducing the time and effort for searching, manipulating, and navigating through the data. However, there are some limitations: it needs a bit more space than the multi-thumb sliders and it is not applicable if there are empty ranges, i.e., if there are values between the selected ranges that are filtered out. Finally, to provide a higher level for the provided information, the total number of objects in each range, a legend is provided next to the RSF Slider as shown in Figure 14 (Slider videos: Al-Zokari, Y, 2020).



Figure 14The dynamic legend showing the number of objects in each range.

## 5. ASSESSMENT OF EASE OF USE

For assessing the ease of use of our RSF slider, we chose three adaptable ranges coded as *H*: high (in red), *M*: Medium (in yellow), and *L*: Low (in green). All movements for changing the ranges are shown in Figure 15. The *cascading effect* helps in reducing the movements of the thumbs. Thus, the “ease of use” of the RSF slider is higher than for the multi-thumb slider. The restriction for the multi-thumb slider is resolved by enabling the left thumb to move and push the right thumbs if they are at the same position. Similarly, the right thumb can move and push the left thumbs. The colored arrows show which thumb should be moved. The numbers show the minimum amount of movements that is needed to change the range. The different levels in light blue show the movements that cannot be performed at the same time. This means, if two movements are in the same level, i.e., the same light blue line that represents the body of the slider, any of the two movements can be performed, i.e., there is no restriction in the order of the movements. Thus, it is more flexible to choose the movement. However, if the movements are not in the same level, then if the movement order is different, the number of movements will increase. Therefore, the number of movements will not be the minimum any more for the same case.

From the root to the leaves of Figure 16, the total number of minimum thumb movements for both the multi-thumb and the RSF slider are counted. The reduction or increase *c* of the number of movements *n* is computed as shown in Equation 2.

$$\begin{aligned}
 n_{MTS} &:= \text{No. of movements of the multi – thumb slider} \\
 n_{ES} &:= \text{No. of movements of the Dynamic Slider} \\
 \text{Equation 2 } c &:= \frac{n_{MTS} - n_{ES}}{n_{MTS}} \\
 ratio &:= \frac{n_{ES}}{n_{MTS}} \\
 LMH \rightarrow H \rightarrow M \rightarrow L: & 5/8 \text{ (a reduction of } c = 37.5\% \text{ movements)} \\
 LMH \rightarrow H \rightarrow LM: & 4/5 \text{ (a reduction of } c = 20\% \text{ movements)} \\
 LMH \rightarrow MH \rightarrow L: & 3/4 \text{ (a reduction of } c = 2\% \text{ movements)}
 \end{aligned}$$

Further, for other domains where the analysis steps do not have the same restriction sequence as in the one of the MCS analysis II context, and different sequences can be performed, such as the steps:  $L \rightarrow LMH \rightarrow L \rightarrow H \rightarrow M \rightarrow MH$ , the reduction difference is even higher (Figure 15):

$$\begin{aligned}
 LM \rightarrow LMH: & 1/7 \text{ (a reduction of } 85.71\% \text{ movements)} \\
 LMH \rightarrow L: & 1/2 \text{ (a reduction of } 50\% \text{ movements)} \\
 L \rightarrow H: & 2/7 \text{ (a reduction of } 71.42\% \text{ movements)} \\
 H \rightarrow M: & \text{(same as previous example): } 3/4 \text{ (a reduction of } 25\% \text{ movements)} \\
 M \rightarrow H: & 2/7 \text{ (a reduction of } 71.42\% \text{ movements)}
 \end{aligned}$$

Additionally, an empirical evaluation was performed by (Al-Zokari, 2016) pp.386. It was not included in this paper due to the limited number of pages available.

## 6. APPLICATIONS

### 6.1. Software Visualization

In the area of software engineering, the static structure of a system is often represented using graphs. Considering object-oriented systems, nodes represent classes or interfaces. Edges can represent different relations of these classes and interfaces (Abuthawabeh & Zeckzer, 2013), Figure 17, proposed and (Abuthawabeh, et al., 2013) evaluated a matrix-based visualization of these graphs. The graphs and their visualizations are frequently used to solve software engineering tasks: reengineering of legacy systems, software understanding, debugging, and construction of systems, to name a few. For solving these tasks, not only the existence of links is important but also their strength. The latter becomes even more important when non-functional relationships are investigated. The strength of the relation is mapped to color saturation. To ease the recognition of differences, only three saturation levels are used. Hence, different strengths are combined into one strength level. The RSF slider is used in their work to determine, which intervals are mapped to which saturation.

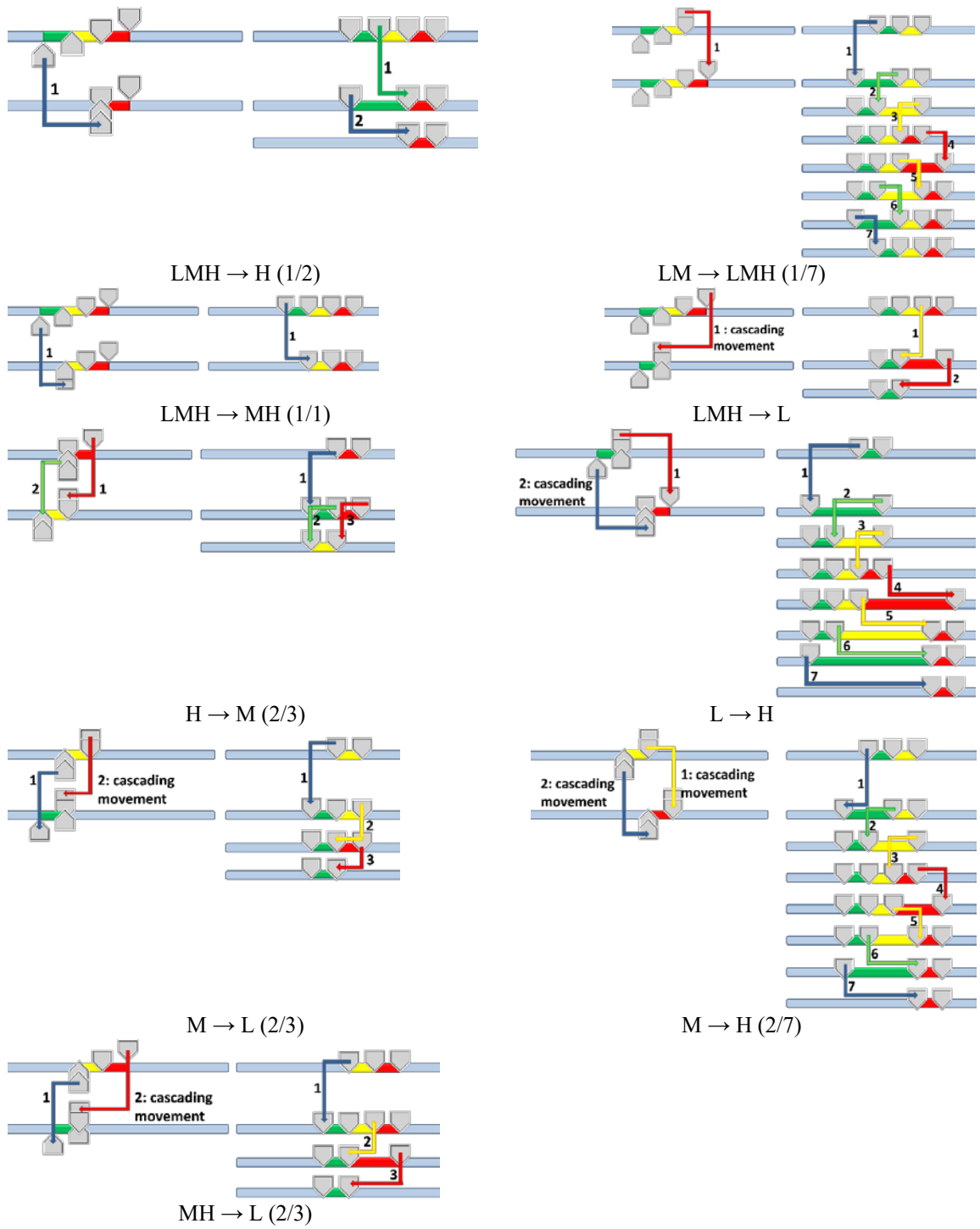


Figure 15 Minimum number of thumb movements for both multi-thumb and Dynamic Sliders. Format (# of movements for Dynamic Slider / # of movements for multi-thumb slider).



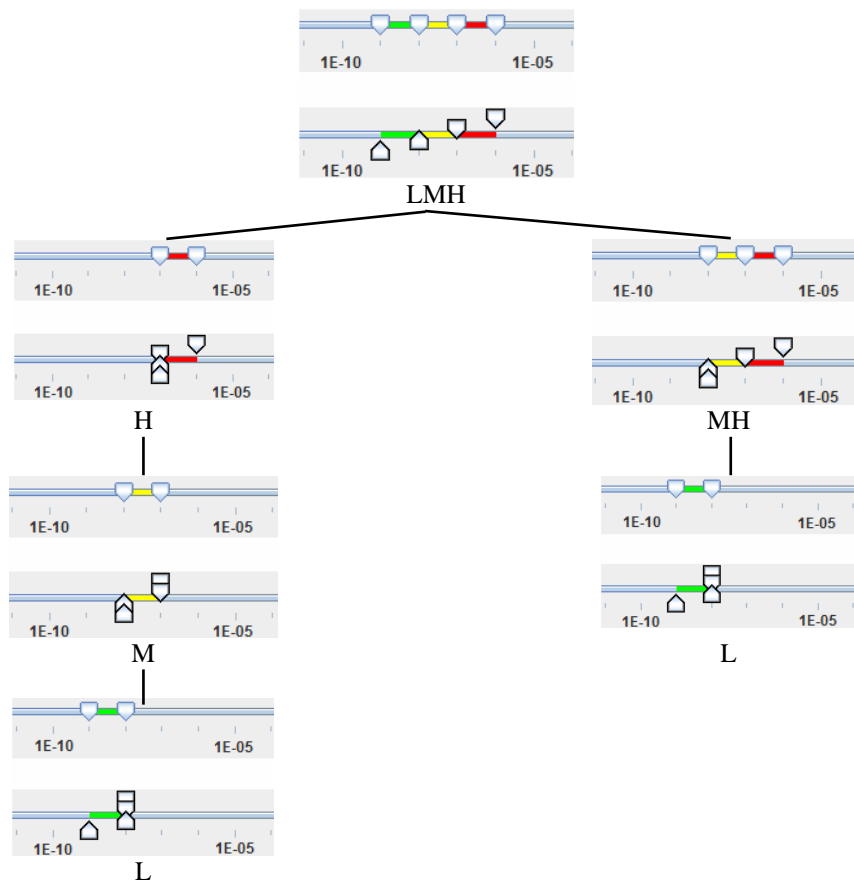


Figure 16 The representations of the multi-thumb (top) slider and the Dynamic Slider (bottom) for different analysis steps starting by user defined FP levels. H: high (red), M: medium (yellow), and L: low (green).

Regarding ‘evolution’ (evolutionary coupling, violet, fourth slider from top), a strength between 1 and 11 is considered to be low (least saturated), a strength between 12 and 22 is considered to be medium (medium saturation), and a strength between 22 and 32 is considered to be high (highest saturation). Here, the links having a strength below 11 are filtered out. In the matrix view (left) a large amount of evolution links is filtered out allowing to analyze links with medium and high strength. This helps the software engineer while analyzing a system. If parts (classes) of the system are commonly changed together, there is a semantic relationship. Therefore, this semantic relationship is important for software understanding and re-engineering and must be considered while implementing new functionality or fixing bugs.

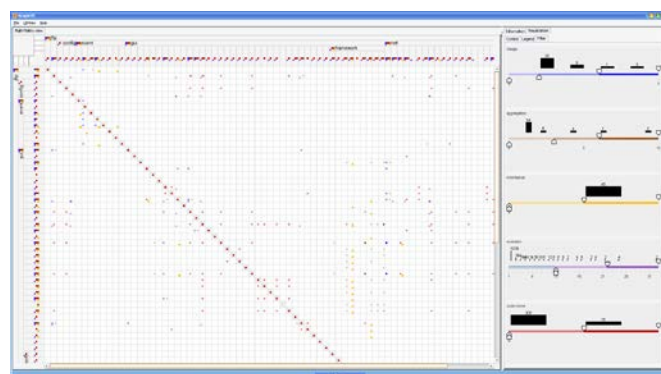


Figure 17 View of a software system. All edge types are visualized with saturations according to their strength. Low evolution scores are filtered out.

## 6.2. Safety

In the area of (embedded) systems engineering, safety analysis is performed to avoid harm of the system to people or the environment (Bozzano & Villaflorita, 2010). Often, the individual values are of minor importance in this field. As they are based on estimations, their order of magnitude is the dominant factor. Moreover, for practical reasons, criticality is often determined by categorizing the numerical values assigned. One possible scheme called ALARP (Melchers, 2001) divides the complete range of failure probabilities into three categories: negligible, tolerable, and unacceptable. Commonly, the colors red (unacceptable), yellow (tolerable), and green (negligible) are used for these categories.

In (Al-Zokari, et al., 2011a), (Al-Zokari, et al., 2011c), and (Al-Zokari, et al., 2012d), a system was presented that displayed so called minimal cutsets (MCSs) based on the ALARP categorization of criticality. It supports to identify the system hazards with the highest failure probability based on MCS analysis. In (Al-Zokari, 2016) pp.497ff the following tasks were identified:

- T-1 Identify all MCSs according to the standard ranges.
- T-2 Identify MCSs with high FP (unacceptable). How many MCSs should be analyzed?
- T-3 Identify MCSs with small size (also called order).
- T-4 The combination of both T-2 and T-3, which is essential for finding unusual MCSs, e.g., those having a low FP and a small size.
- T-5 How safe is the hazard? Provide an overview over the safety of the hazard.
- T-6 How many MCSs should be analyzed when unacceptable MCSs are already solved (tolerable and/or negligible)? Identify them.
- T-7 Are there any MCS outside the system's standard analysis or the current investigation range (above unacceptable and/or below negligible)?
- T-8 How many MCSs with FPs lower than a certain value or in a certain range should be analyzed? Identify them.
- T-9 Identify the MCSs in the ranges below the high FP range (tolerable, negligible, and/or below negligible). How many should be analyzed?

Almost all safety analysts have little knowledge of visualization or interaction techniques. Therefore, if interaction (e.g., filtering) is required to achieve their tasks, the interaction technique should be easy to use, and helping to increase their productivity. The RSF slider presented in Section 4 is useful for this application. It supports to perform the safety tasks T-1 – T-2 and T-5 – T-9. Using the RSF slider, the user can easily use it for choosing those parts of the data range that she wants to filter (remove), as well as choosing which part of the data range is considered to be negligible (low), tolerable (medium), or unacceptable (high). As these ranges are supposed to be adjacent, no empty ranges can occur.

Figure 18 shows the two important parts of the system: the RSF slider and the visualization of the MCSs. The safety analyst sees that there are 6 MCSs in the high failure probability range that have to be analyzed first, 276 in the medium range, and 6 in the low range, so Tasks T-1 and T-2 are answered. This allows to answer the questions raised by Task T-7 (No) and Task T-6 (276). The answer to the question raised in Task T-5 is that the hazard is unsafe because there are 6 MCSs above the unacceptable range, but if these MCSs are solved the system will be tolerable. To accomplish Task T-8, e.g., for a failure probability lower than  $10^{-18}$  would be 6 MCSs. For Task T-9, the answer depends on the amount of time and resources available. For example, if there is a short notice request for MCS analysis, e.g., 1 hour, the number of MCSs that have to be (can be) analyzed are 2 (the single points of failure). If there is a more time available (e.g., 3 hours), the number is 6 (the other 4 from the red range), etc.

## 7. SUMMARY AND FUTURE WORK

In this paper, we presented the Range-Selection-and-Filtering (RSF) slider that supports the users getting multiple levels of overview of the data being analyzed and filtering according to multiple ranges. We combined the features of multiple slider types, i.e., histogram, interactive legends, standard, and multi-thumb sliders. Moreover, we increased its usefulness and its ease of use by repositioning the thumbs to overcome the overlapping of thumbs and adapting the orientation of the thumbs' icons and introducing the cascading effect concept and applying it. This led to reducing the amount of interaction.

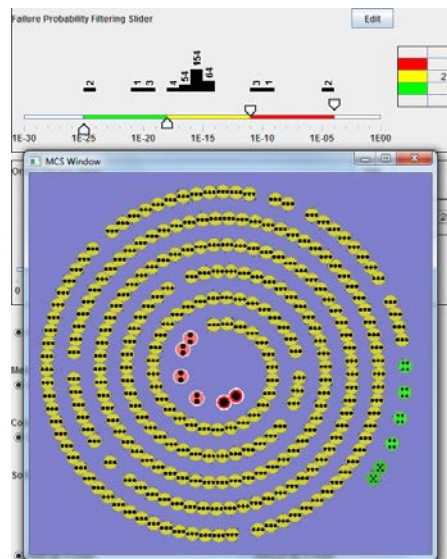


Figure 18 Slider with histogram and the data being analyzed (data size 288 MCSs).

We validated our design against the different slider types. From our case studies, we found that using the RSF slider is direct and no unnecessary thumb moves are required. Additionally, our design supports different color vision deficiencies by positioning the thumbs at a logical level that indicates the thumb's value. However, there are some limitations in this design, as it takes a little more space than the multi-thumb slider, and no empty ranges are possible. Nevertheless, this case is not a requirement in Minimal Cut Sets analyses, safety domain nor in Software Engineering. However, for future work having the possibility to enable empty ranges should be considered for other domains that require such a property. In the future, we need to perform an evaluation of our design to compare it to other sliders regarding usefulness and usability. Nevertheless, an empirical evaluation of our RSF slider was performed (Al-Zokari, 2016) pp.386, but we did not include it in this paper due to the limited number of pages. The source code and executable files are available at (Al-Zokari, Y, 2020. Slider Source Code in GitLab).

## ACKNOWLEDGEMENT

This work was supported partially by DAAD, IRTG 1131, and BMBF. We would like to thank our safety expert collaborator Mr Mohammed. Hibti from EDF "Electricite De France" for his time and collaboration.

## REFERENCES

- Abuthawabeh, A., Beck, F., Zeckzer, D. & Diehl, S., 2013. Finding Structures in Multi-Type Code Couplings with Node-Link and Matrix Visualizations. In 1<sup>st</sup> IEEE Working Conference on Software Visualization 2013 (VISSOFT 2013).
- Abuthawabeh, A. & Zeckzer, D., 2013. IMMV: An Interactive Multi-Matrix Visualization for Program Comprehension. In 1<sup>st</sup> IEEE Working Conference on Software Visualization 2013 (VISSOFT 2013).
- Ahlberg, C., 1996. Spotfire: an information exploration environment. SIGMOD Rec. 25.
- Ahlberg, C. & Shneiderman, B., 1994. The alphaslider: a compact and rapid selector. In Proc. of the SIGCHI conference on Human factors in computing systems: celebrating interdependence (New York, NY, USA, 1994), CHI '94, ACM, p. 365–371.
- Ahlberg, C. & Shneiderman, B., 1994. Visual information seeking using the FilmFinder. In Conference companion on Human factors in computing systems (New York, NY, USA, 1994), CHI '94, ACM, p. 433–434.
- Ahlberg, C., Williamson, C. & Shneiderman, B., 1992. Dynamic queries for information exploration: an implementation and evaluation. In Proc. of the SIGCHI conference on Human factors in computing systems (1992), ACM, p. 619–626.

- Al-Zokari, Y. I. et al., 2011. CakES: Cake Metaphor for Analyzing Safety Issues of Embedded Systems. In: H. Hagen, ed. Scientific Visualization: Interactions, Features, Metaphors. Dagstuhl(Germany): Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, p. 1–16.
- Al-Zokari, Y. I., Schneider, D., Zeckzer, D. & Hagen, H., 2011. An Enhanced Slider for Safety Analysis. In IEEE VisWeek Poster (2011).
- Al-Zokari, Y. I. et al., 2011. Enhanced CakES representing Safety Analysis results of Embedded Systems. In Federated Conference on Computer Science and Information Systems (FedCSIS) (2011).
- Al-Zokari, Y. I. et al., 2012. Towards Advanced Visualization and Interaction Techniques for Fault Tree Analyses Comparing existing methods and tools. In PSAM11 & ESREL 2012 Conference (2012).
- Al-Zokari, Y., 2020. Slider Videos. [https://www.youtube.com/playlist?list=PLZ9SRRNo8TbbJa8cjxdkuYjwshlH\\_IGI](https://www.youtube.com/playlist?list=PLZ9SRRNo8TbbJa8cjxdkuYjwshlH_IGI) .
- Al-Zokari, Y., 2020. Slider Source Code in GitLab. <https://gitlab.com/yaso2com/rsf-slider> .
- Borland, D. & Taylor, R. M., 2007. Rainbow Color Map (Still) Considered Harmful. Computer Graphics and Applications, IEEE, 3, Volume 27, pp. 14-17.
- Bozzano, M. & Villaflorida, A., 2010. Design and Safety Assessment of Critical Systems. CRC Press (Taylor and Francis), an Auerbach Book.
- Card, S. K. & Mackinlay, J., 1997. The structure of the information visualization design space. In Proc. of the 1997 IEEE Symp. on Information Visualization (InfoVis '97) (Washington, DC, USA, 1997), IEEE Computer Society, p. 92–99.
- Dang, G., North, C. & Shneiderman, B., 2001. Dynamic Queries and Brushing on Choropleth Maps. In Proceedings of the Fifth International Conference on Information Visualisation (2001), IEEE Computer Society, p. 757–764.
- Eick, S. G., 1994. Data visualization sliders. In Proc. Of the 7th annual ACM symposium on User interface software and technology (1994), ACM, p. 119–120.
- Fekete, J.-D., 2004. The InfoVis Toolkit. In Proc. of the IEEE Symposium on Information Visualization (Washington, DC, USA, 2004), IEEE Computer Society, p. 167–174.
- Harris, C., Allen, R., Plaisant, C. & Shneiderman, B., 1999. Temporal Visualization for Legal Case Histories. Tech. Rep. Vol. 36, ASIS'99 Proceedings of the 62nd Annual Meeting of the American Society for Information Sciences, 1999.
- HCI Lab, U.S. Bureau of Census, n.d. User Interfaces for the U.S. Bureau of Census Online Survey Interfaces and Data Visualization. <http://www.cs.umd.edu/hcil/census/>; Online; last access 29-Mar-2011.
- Heer, J., Card, S. K. & Landay, J. A., 2005. Prefuse: a toolkit for interactive information visualization. In Proc. of the SIGCHI conference on Human factors in computing systems (2005), ACM, p. 421–430.
- Henry, N. R., Lee, B. & Plaisant, C., 2010. Understanding Interactive Legends: a Comparative Evaluation with Standard Widgets. Computer Graphics Forum, Volume 29, p. 1193–1202.
- Li, Q. et al., 2003. Dynamic query sliders vs. brushing histograms. In CHI'03 extended abstracts on Human factors in computing systems (New York, NY, USA, 2003), CHI EA '03, ACM, p. 834–835.
- Melchers, R. E., 2001. On the ALARP approach to risk management. Reliability Engineering & System Safety, Volume 71, 2, pp. 201-208.
- Plaisant, C. & Laboratory, H. C. I., 1993. Facilitating Data Exploration: Dynamic Queries On A Health Statistics Map. In Proc. of the Government Statistics Section, Annual Meeting of the American Statistical Assoc. (1993), p. 18–23.
- Savive Pty Ltd, Safety-Critical. <http://www.savive.com/inform/safetycritical.html>; Online; accessed 31-Mar-2011.
- Shneiderman, B., Feldman, D., Rose, A. & Grau, X. F., 2000. Visualizing digital library search results with categorical and hierarchical axes. In Proc. of the fifth ACM conference on Digital libraries (2000), ACM, p. 57–66.
- Swing, Swing Examples. [http://en.pudn.com/downloads3/sourcecode/java/detail6037\\_en.html](http://en.pudn.com/downloads3/sourcecode/java/detail6037_en.html); Online; last access 06-June-2011.
- Tanin, E., Beigel, R. & Shneiderman, B., 1996. Incremental data structures and algorithms for dynamic query interfaces. SIGMOD Rec. 25, p. 21–24.
- Tweedie, L., Spence, R., Dawkes, H. & Su, H., 1996. Externalising abstract mathematical models. In Proc. of the SIGCHI conference on Human factors in computing systems: common ground (1996), ACM, p. 406–ff.
- Ware, C., 2004. Information Visualization: Perception for Design. Morgan Kaufmann Publishers Inc..
- Williamson, C., Shneiderman, B. & Ablex, E., 1992. The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. In Proceedings of the 15<sup>th</sup> annual international ACM SIGIR conference on Research and development in information retrieval (1992), no. Vol. 36 in SIGIR '92, pp. 21-24,338-346.