This supplementary document provides an overview of the datasets that we used in our evaluation (Section 1), a detailed description of all metrics (Section 2), and running time results of both our algorithm and the industry standard *dot* for all datasets (Section 3). We also describe the qualitative user study in detail (Section 4). Finally, we compare a number of layout results of our approach and *dot* (Section 5).

1. Datasets

Name	Nodes	Edges	Cases	Variations	Description
BPI2012	36	188	23967	4366	Dataset of the second international business process intelligence challenge [3]
BPI2013	13	81	30239	1838	Dataset of the third international business process intelligence challenge [4]
BPI2014	39	772	88502	21000	Dataset of the fourth international business process intelligence challenge [5]
BPI2017	48	257	73364	16263	Dataset of the seventh international business process intelligence challenge [6]
BPI2017RCA	26	167	42995	2836	A simplified version of the BPI2017 dataset
BPI2018	41	594	466141	26104	Dataset of the eighth international business process intelligence challenge [7]
Electricity	22	300	144339	10874	Dataset containing process data of an electricity company
Hospital	18	144	100000	780	Dataset containing process data of a hospital
Insurance	13	87	99999	22149	Dataset containing process data of an insurance company
Invoices	15	48	25743	101	Our running example
P2P	50	615	80017	5807	Dataset containing process data of a purchase to pay process
Road	11	70	150370	171	Dataset of an information system managing road traffic fines
Sepsis	16	155	3383	806	Dataset containing process data of how sepsis (a potentially life-threatening complication of an infection) is handled

Table 1: *The datasets used to test the algorithms.*

2. Metrics

This section contains detailed descriptions of the used quality and stability metrics. Most of these metrics take the weight of edges into account because we deem it important to optimize graph aesthetics for the edges with highest weight. Intuitively, edges with highest weight represent the most prominent process behaviour. Also, computing exact values for the edges (such as intersections or edge length) is quite expensive because the edges are drawn as splines. Therefore, consider that an edge e is drawn as a sequence of spline segments, which we denote as $segments(e)$. Every segment $s \in segments(e)$ contains four points (coordinates): a start point p_s at which the spline segment starts, two control points p_{c1} and p_{c2} that define the shape of the spline segment, and an end point p_e at which the spline segment ends. We use an approximation for the edge e, which consists of approximations of the spline segments $s \in segments(e)$. An approximation of a spline segment is the straight line $line(s)$ between p_s and p_e. Moreover, the euclidean distance between p_s and p_e is an approximation of the actual length of s and is denoted as $length(s)$. Consequently, the approximate length of an edge e is defined as:

$$length(e) = \sum_{s \in segments(e)} length(s)$$

2.1. Quality Metrics

Running Time

The running time of the layout algorithm is not really an aesthetic criterion since it does not relate to the computed graph layout. Nevertheless, a faster algorithm is obviously more desirable to the user since this reduces waiting times. Therefore, the quality metric $QM_{time}(G)$ measures the time (in milliseconds) the layout algorithm required to compute the graph layout for G.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

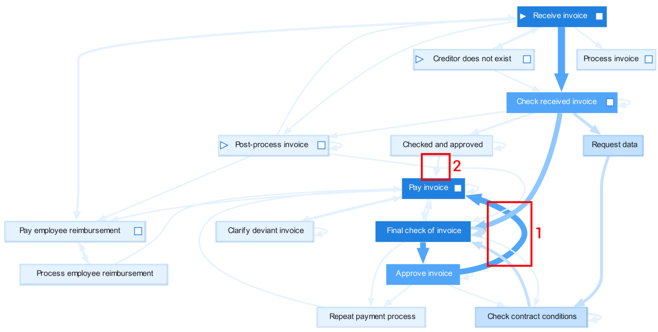


Figure 1: *A layout computed by dot for the invoices data set. Clearly, the edge crossings in red rectangle 1 are more salient than those in red rectangle 2.*

Edge Crossings

The number of edge crossings in a graph layout has a significant effect on the readability [PCJ97, Pur00]. Therefore, we define the following function that computes the 'cost' of the edge crossing(s) between two edges.

$$cross(e_1,e_2) = \sum_{(s_1,s_2)\in segments(e_1)\times segments(e_2)} cr(e_1,e_2,s_1,s_2)$$

Where $cr(e_1,e_2,s_1,s_2)$ is defined as follows:

$$cr(e_1,e_2,s_1,s_2) = \begin{cases} 0 & \text{line(s\_1) and line(s\_2) do not intersect} \\ weight(e_1)\cdot weight(e_2) & \text{line(s\_1) and line(s\_2) intersect} \end{cases}$$

Since edges of higher weight are drawn more salient, edge crossings of high weight edges are also more salient. For example, in Figure 1, the crossings in red rectangle 1 are more salient than the edges crossing in red rectangle 2. Therefore, to quantify that we consider high weight edge crossings as worse, we multiply the weights of e_1 and e_2. Let $pairs(E)$ be the set of unique edge pairs in a graph G. The edge crossings quality metric is defined as follows:

$$QM_{crossings}(G) = \sum_{(e_1,e_2)\in pairs(E)} cross(e_1,e_2)$$

Average Edge Length

Short edges are easier to follow [GKNV93]. Moreover, we deem it important to have short high weight edges. Therefore, the average edge length quality metric, as defined below, takes the weight of an edge into account.

$$QM_{avg\_length}(G) = \frac{1}{|E|}\sum_{e\in E} length(e)\cdot weight(e)$$

Edge Bends

Edge bends negatively affect graph readability [PCJ97, Pur00]. Our approach towards computing the number of edge bends is inspired by the work of Bridgeman et al. [BT98]. For an edge segment s, we define direction $dir(s) \in \{N,E,S,W\}$ (North, East, South, West) as the most prominent direction of $line(s)$. For example, in Figure 2 the direction of every edge segment is illustrated. As we can see, the shape of an edge can be described as a sequence of directions. The concatenation of these directions results in a shape string [BT98] $string(e)$. For example, in Figure 2, the shape string of the rightmost edge equals $string(e) = \text{ESSW}$. Consequently, the number of bends $bends(e)$ in an edge equals the number of times there is a change in segment direction. For example, in Figure 2, the bends are indicated by black circles. The edge bends quality metric is defined as follows:

$$QM_{bends}(G) = \sum_{e\in E} bends(e)\cdot weight(e)$$

Since we prefer high weight edges to be straight, we multiply by the edge weight.

Back Edges

Since we are drawing graphs vertically in a top-down manner, we want to have as many forward edges as possible. The amount of forward

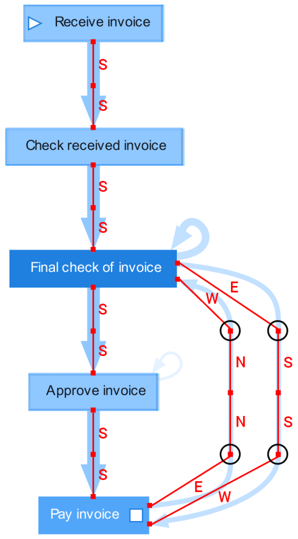


Figure 2: *Illustration of the directions of every edge segment. Red squares indicate the start and end points of edge segments. Red lines illustrate the line $line(s)$ of an edge segment s. The direction $dir(s)$ value of every segment is indicated next to the respective red line. Moreover, the red squares encircled in black indicate edge bends. In total, there are 4 edge bends in this graph layout.*

edges directly relates to the consistency of 'flow direction' in the graph layout. Therefore, let $backedges(G)$ be the set of back edges in graph layout G. Then, the back edge quality metric is defined as follows:

$$QM_{back\_edges}(G) = \sum_{e \in backedges(G)} weight(e)$$

Since we want high weight edges to be forward edges, we sum up the weights of the back edges instead of simply counting the number of back edges.

Flow Direction

The back edge quality metric relates to the consistency of the flow direction in a graph layout. However, since this metric does not take the actual shape of an edge into account, it does not properly describe the flow in terms of what the user sees on the screen. For example, a forward edge e, which, intuitively, is drawn downward, can have, for all segments $s \in segments(e)$, a direction $dir(s) = \text{E}$. Therefore, we introduce a quality metric, similar to the 'direction of a model' metric of Bernstein et al. [BS15], that measures the flow direction of the edges in a graph layout G. Let $north(G)$, $east(G)$, $south(G)$, $west(G)$, be the number of edge segments s in G that have $dir(s)$ value N, E, S, W respectively. Intuitively, since we are drawing graphs vertically in a top-down manner, all $north(G)$ and $south(G)$ edge segments adhere to the vertical drawing direction while $east(G)$ and $west(G)$ edge segments 'break the flow'. Therefore, the flow direction quality metric is defined as follows:

$$QM_{flow}(G) = \frac{east(G) + west(G)}{north(G) + south(G)}$$

Note that the lower $QM_{flow}(G)$, the better the flow direction adheres to the (vertical) drawing direction of the graph layout. Also, we do not distinguish between $north(G)$ and $south(G)$ because this is already captured by the $QM_{back\_edges}(G)$ quality metric.

Area

The quality metric $QM_{area}(G)$ computes the area required to draw the graph layout (including edge labels). Preferably, in order to fit a graph layout on the screen of the user, the area of a graph layout is a small as possible.

Discarded Metrics

Some quality aesthetics, as discussed below, found in the literature, are not used for different reasons.

Some layout algorithms aim to make a graph layout as symmetrical as possible. Since a process does not necessarily contain symmetries, we do not include this aesthetic.

Other layout algorithms, in order to fit the resulting layout on the screen, aim to compute a graph layout that has an aspect ratio of 1:1. Since this aesthetic can contradict quite some other aesthetic criteria (flow direction, back edges, edge bends), we do not consider this aesthetic.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

Other alternative aesthetics related to the length of edges exist. We could, for example, compute the minimum/maximum edge length in a graph layout. However, since the average edge length relates to all edges, and not just to the shortest/longest, we decided to use the average of edge lengths.

2.2. Stability Metrics

Measuring graph stability essentially comes down to measuring the degree of change between two graphs. When the degree of change between graphs is small, there is a higher degree of stability. In the field of graph drawing, the problem of how to measure change (or similarity) has received quite some attention [BT98, LMR98, BT00, DG02]. After investigating the stability metrics proposed in these works and after considering our mental map definition [MELS95], we constructed the list of stability metrics listed in the sections below. We denote the euclidean distance between two points (coordinates) by $dist(p_1, p_2)$. For a stability metric $SM_\lambda(G, G')$, the equivalent of a node $n_i \in V$ in V' is denoted by n'_i. For example, the distance between the old and new position of a node n equals $dist(pos(n), pos(n'))$. Additionally, difference metrics only measure based on nodes or edges that are in both layouts, i.e. the sets $V \cap V'$ and $E \cap E'$. Moreover, let $pairs(V \cap V')$ and $pairs(E \cap E')$ be the sets of unique node/edge pairs in $V \cap V'$ and $E \cap E'$ respectively.

Relative Euclidean
The relative euclidean stability metric measures, for all node pairs present in both layouts, the change in relative distance. This metric is defined as follows:

$$SM_{rel\_eucl}(G, G') = \sum_{(n_i, n_j) \in pairs(V \cap V')} |\, dist(pos(n_i), pos(n_j)) - dist(pos(n'_i), pos(n'_j))\,|$$

By taking the absolute value, we make sure the metric is always positive.

Hausdorff
The hausdorff distance is a standard metric used to measure the match between two point sets [BT98]. A disadvantage, however, is the fact that the standard hausdorff metric does not take point labels into account. More specifically, every point in the point set is seen as a coordinate and not as a unique entity. Consequently, swapping the position of two points in the point set does not change the point set. Since we do have 'labeled' points (nodes), this standard metric does not work for us. Therefore, we consider an adapted version of the hausdorff distance metric [BT00] (that does take point labels into account) which is defined as follows:

$$SM_{hausdorff}(G, G') = \max_{n \in V \cap V'} dist(pos(n), pos(n'))$$

Orthogonal
The orthogonal ordering mental map model [MELS95] states that a layout adjustment should preserve the direction of node n to node m for each pair of nodes n and m. In order to quantify this, we define the following stability metric:

$$SM_{orthogonal}(G, G') = \sum_{(n_i, n_j) \in pairs(V \cap V')} angle(pos(n_j) - pos(n_i), pos(n'_j) - pos(n'_i))$$

The *angle* function computes the smallest positive angle between the two provided vectors. As we can see, we provide the vectors for the old and new layout between the nodes in the node pair. Therefore, we essentially use the change in angle to express the change in direction between two nodes.

Epsilon-Cluster
The proximity mental map model [MELS95] states that nodes which are close together should remain close together after a change. An ε-cluster for a node n_i is the set of nodes n_j such that $dist(pos(n_i), pos(n_j)) \leq \varepsilon$. In this work, ε is defined as the largest nearest neighbor distance, i.e. [ELMS91]:

$$\varepsilon = \max_{n_i} \min_{n_j \neq n_i} dist(pos(n_i), pos(n_j))$$

The ε-cluster stability metric measures how the ε-cluster of node n_i compares to that of node n'_i. Note that ε can be different for G and G', denoted by ε and ε' respectively. Consequently, we define the following sets:

$$\varepsilon(G) = \{(n_i, n_j) \in pairs(V \cap V') \,|\, dist(pos(n_i), pos(n_j)) \leq \varepsilon\}$$
$$\varepsilon(G') = \{(n_i, n_j) \in pairs(V \cap V') \,|\, dist(pos(n'_i), pos(n'_j)) \leq \varepsilon'\}$$

The ε-cluster stability metric is then defined as:

$$SM_{cluster}(G, G') = 1 - \frac{|\,\varepsilon(G) \cap \varepsilon(G')\,|}{|\,\varepsilon(G) \cup \varepsilon(G')\,|}$$

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

Observe that when the ε-clusters of nodes do not change, this metric returns 0.

Edge Shape

While most stability metrics measure change in terms of the positions of nodes, Bridgeman et al. [BT00] propose a metric that measures the change in the shape of the edges. This metric is defined as follows:

$$SM_{edge\_shape}(G,G') = \sum_{e \in E \cap E'} edit(string(e), string(e'))$$

The *edit* function computes the levenshtein distance [Lev66] between the two edge shape strings. Intuitively, a large edit distance implies that the shape of an edge changed significantly.

Discarded Metrics

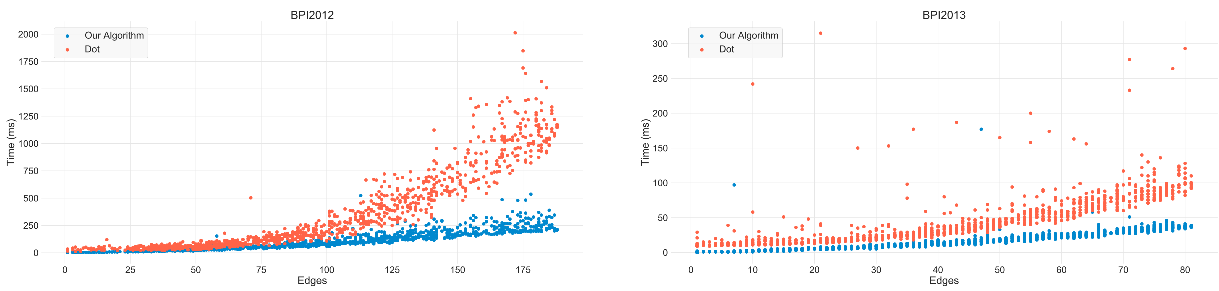
Some stability metrics found in the literature, as discussed below, are not used for different reasons.

The absolute euclidean distance metric is an intuitive and commonly used metric [BT98, LMR98, BT00, DG02]. However, because it is sensitive to global layout translations (which are visually not observable), we exclude this metric. For the same reason, the nearest neighbor between metric, as introduced by Bridgeman et al. [BT98], is also not considered.

An alternative orthogonal ordering metric is the one proposed by Diehl et al. [DG02]. In this metric, the amount of node pairs that change their relative vertical or horizontal direction is measured. Therefore, this metric is essentially a discrete variant of $SM_{orthogonal}(G,G')$. Consequently, it is less accurate.

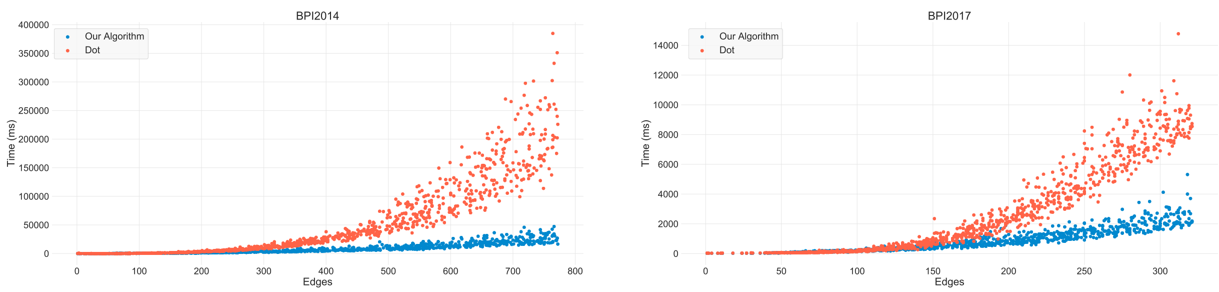
Bridgeman et al. [BT98] introduce the nearest neighbor within metric to express the change in node proximity. We implemented this metric and found it to be too sensitive. Due to rounding issues, the coordinate of a node could change by 1 unit. Consequently, nearest neighbor information could change while no change was visually observable.

3. Running Time Results



(a) (b)

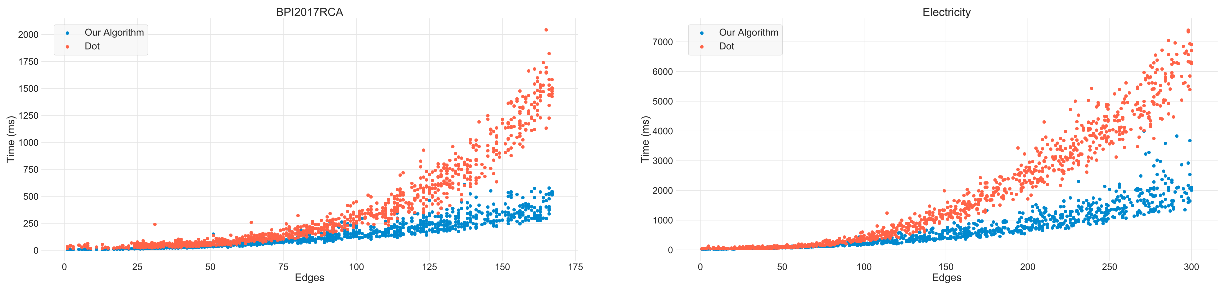
Figure 3: *(a) BPI2012. (b) BPI2013.*



(a) (b)

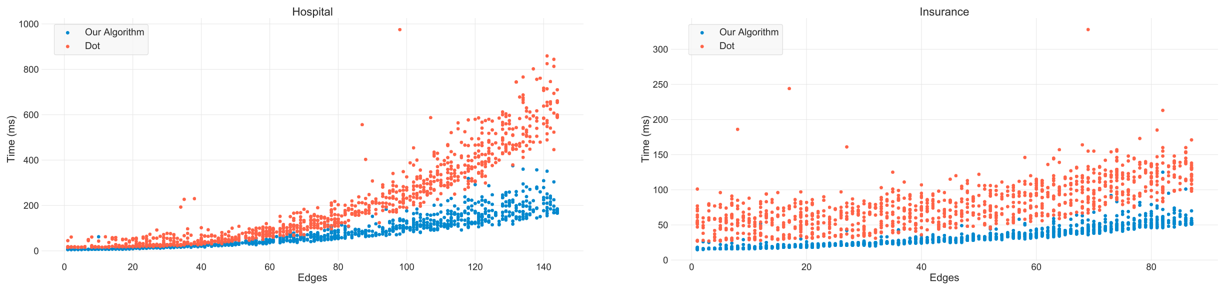
Figure 4: *(a) BPI2014. (b) BPI2017.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.



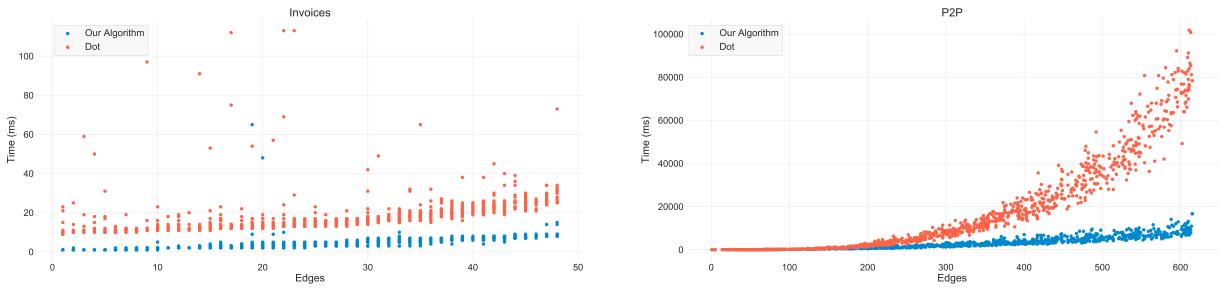
(a) (b)

Figure 5: *(a) BPI2017RCA. (b) Electricity.*



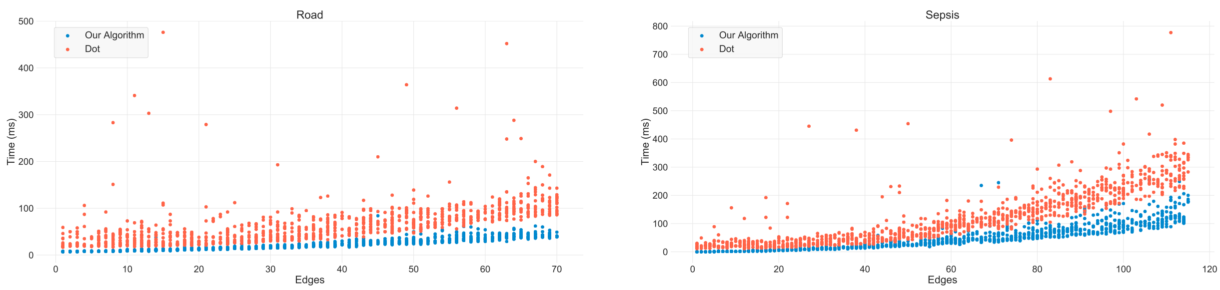
(a) (b)

Figure 6: *(a) Hospital. (b) Insurance.*



(a) (b)

Figure 7: *(a) Invoices. (b) P2P.*



(a) (b)

Figure 8: *(a) Road. (b) Sepsis.*

4. Qualitative Evaluation

To determine which algorithm is preferred by actual users, we performed a user study in which 14 process mining experts participated. In the user study, participants have to work with the process graph dashboard in order to answer 10 process-related questions about two datasets (Road and Insurance, see Section 1 in this document). These dataset questions were created after interviews with 3 process mining experts and are designed to trigger typical process analysis tasks. Participants have to answer the dataset questions three times, each time using a different algorithm (A: Our algorithm, B: *Dot* with animation, or C: *Dot* without animation). The order of the algorithms is randomized among the participants. Note that, after answering the dataset questions the first time, the participant already (thinks he/she) knows the answer to every question. Nevertheless, we ask the participants to re-perform the steps taken to answer the questions. The goal is to let the participant work with every algorithm. After answering all 10 dataset questions three times, the participants are asked to answer a number of evaluation questions. Section 4.1 presents the experiment forms (containing the evaluation questions) that were provided to the participants during the experiment. The 'answer tables', as mentioned in Section 4.1, are not included because they are simple tables in which the participants had to fill in their answers.

4.1. Experiment forms

Hi there!

In this user study you will be working with the process graph dashboard in order to answer questions about two different datasets: a dataset (named ROAD) of an information system managing road traffic fines and a dataset (named INSURANCE) containing process data of an insurance company. In total, there are three different 'algorithms' (named A, B, and C) that 'run' the process graph dashboard. In order to identify which algorithm you prefer the most, you will answer the questions three times, each time using a different algorithm. Note that, after answering the questions the first time, you already (think you) know the answer to every question. Nevertheless, we want you to reperform the steps you took to answer the questions. The goal is to let you work with every algorithm. After answering all the questions using each of the algorithms, there will be some questions about your experience.

Note, you are only allowed to use the process graph dashboard. No other dashboards may be used. Additionally, you may not add any extra filters. All required filters are already present.

The questions for the two datasets are listed below. *Italic* text is used to indicate filter values and underlined text is used to indicate activity names.

ROAD:

- **R1:** which sequence of activities appears to be the most frequent/important?
- **R2:** are all of these most frequent/important activities executed for *Jurisdiction related* activity types?
- **R3:** are all of these most frequent/important activities executed for *Fine related* activity types?
- **R4:** for the years after *2005* (not including 2005). In which year is Payment directly followed by Payment the most?
- **R5:** which activity is executed in *2007* but not in *2008*?

INSURANCE:

- **I1:** which sequence of activities appears to be the most frequent/important?
- **I2:** which activities occur in *2017* but not in *2018*?
- **I3:** for which age group is Re-assessment directly followed by Payment the most?
- **I4:** at which office is Correct payment directly followed by Correct payment the most?
- **I5:** at this office, for which age group is Correct payment directly followed by Correct payment the most?

In the opened ProcessGold application, you should see the following favorites in arbitrary order:

- ROAD A
- ROAD B
- ROAD C
- INSURANCE A
- INSURANCE B
- INSURANCE C

Each of these favorites links to a process graph dashboard that visualizes one of the datasets (ROAD or INSURANCE) using one of the algorithms (A, B, or C). You should open the favorites one by one from top to bottom. For every favorite that you open, you answer the questions for the respective dataset. Answers to the questions should be written in the provided answer tables (make sure you use the table that belongs to the algorithm you are using).

After answering the questions using all three algorithms, please answer the questions in the "User study evaluation table" (Table 2). After answering these questions, you are done with the experiment.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

User study evaluation table		
ID	**Question**	**Answer**
Q1	In general, which algorithm do you prefer the most?	A / B / C / No preference
Q2	Why?	
Q3	For which algorithm did you find the processes to be the most readable/easiest to follow/understandable?	A / B / C / No preference
Q4	Why?	
Q5	Your interaction with the process graph dashboard changed the process being visualized. For which algorithm was this change the easiest to follow?	A / B / C / No preference
Q6	Why?	
Q7	Algorithms A and B use animation. Did the animation help you answer the questions?	Yes / No / No preference

Table 2: *User study evaluation questions. Encircle/fill in your answer.*

Thank you for your participation!

4.2. Results

In total, 14 process mining experts (13 male and 1 female) participated in the user study. All participants were already familiar with the process graph dashboard.

4.2.1. Dataset questions

In our user study, the dataset questions are only used as a means to make the participants work with the different algorithms. Also, not all dataset questions have a single correct answer. Therefore, we deem the (correctness of the) answers to these dataset questions as less relevant. We are more interested in the answers to the evaluation questions. Nevertheless, we shortly discuss the answers to the dataset questions below.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

Figure 9 contains an overview of the answers given to the questions about the road dataset (except R1). As we can see, most answers are correct. During the user test, we noticed that most incorrect answers were given due to misunderstanding the question, or because the participant simply wrote down something different than what he/she actually intended. For example, we observed one participant who concluded that 2007 is the correct answer for R4. Nevertheless, he/she wrote down 2008 instead.

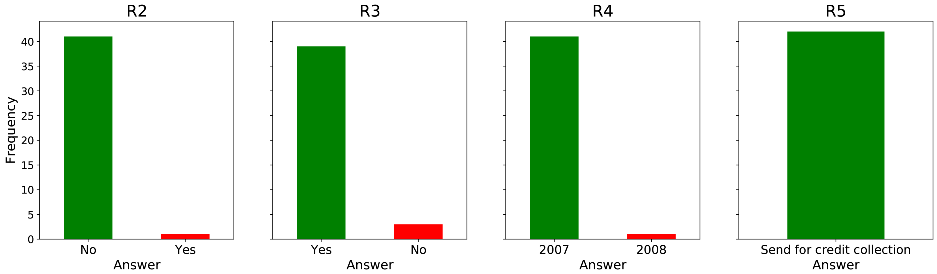


Figure 9: *An overview of the answers given to the questions about the road dataset (except for R1, see Figure 11). Green bars indicate correct answers while red bars indicate incorrect answers.*

Figure 10 contains an overview of the answers given to the questions about the insurance dataset (except I1 and I2). As we can see, similar to the road questions, most answers are correct. Finally, Figure 11 contains the answers to R1, I1, and I2. For R1 and I1, the bars are colored blue because R1 and I1 do not necessarily have a single correct answer. For R1, we can see that most participants identified the same sequence of activities. For I1, on the other hand, there is a lot of variation in which sequence is identified. This is mainly because in the insurance dataset, there is no clear 'main path'. Finally, all participants correctly answered question I2.

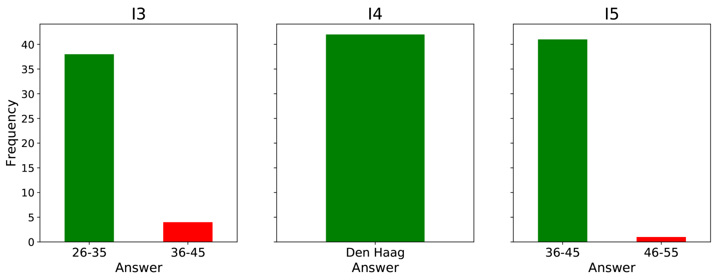


Figure 10: *An overview of the answers given to the questions about the insurance dataset (except for I1 and I2, see Figure 11). Green bars indicate correct answers while red bars indicate incorrect answers.*

4.2.2. Evaluation questions

An overview of the answers given to the evaluation questions, is provided in Figure 12. As we can see, our algorithm (A) performs significantly better than *dot* (B and C). For Q1 and Q3, all participants prefer our algorithm. To verify these results, we run a Chi-Squared test, which returns a p-value of 0.0001828, indicating that with a significance value of $\alpha = 0.05$, the null-hypothesis can be rejected, i.e., there is significant evidence that these results are not obtained by chance. For Q5, there is only one participant who prefers algorithm B. Again, we run a Chi-Squared test and obtain a p-value of 0.0013406, which again results in the rejection of the null-hypothesis (with a significance value of $\alpha = 0.05$). After investigation, we find that this single participant prefers algorithm B mainly due to question I2. For this question, the participants have to identify a set of activities that is present in 2017, but not in 2018. It happened (by chance), that in the layout computed by *dot* (algorithm B), this set of activities is nicely clustered together, and thereby easily identifiable.

Related to animation, all participants prefer animation over no animation (Q7) (and thus the Chi-Squared results for Q1 and Q3 also apply to Q7), which is in line with previous animation research [BB99, SIG07, ZKS11, BPF14, AP16]. As a side note, however, a single participant indicated that he/she only prefers animation for our algorithm (A). For *dot* (B), he/she found the animation to be distracting because a lot of nodes/edges are modified, i.e., the lack of stability results in distracting animations.

Questions Q2, Q4, and Q6 are more difficult to analyze because they are open questions. Nevertheless, answers to these questions indicate that, in general, participants prefer our algorithm because it computes graph layouts where the 'main path' is positioned in the center and because edges are more straight and therefore easier to follow. Additionally, participants state that the stability and animation combination makes it easy to follow changes. All in all, given these results, we can conclude that users prefer our algorithm over *dot*.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

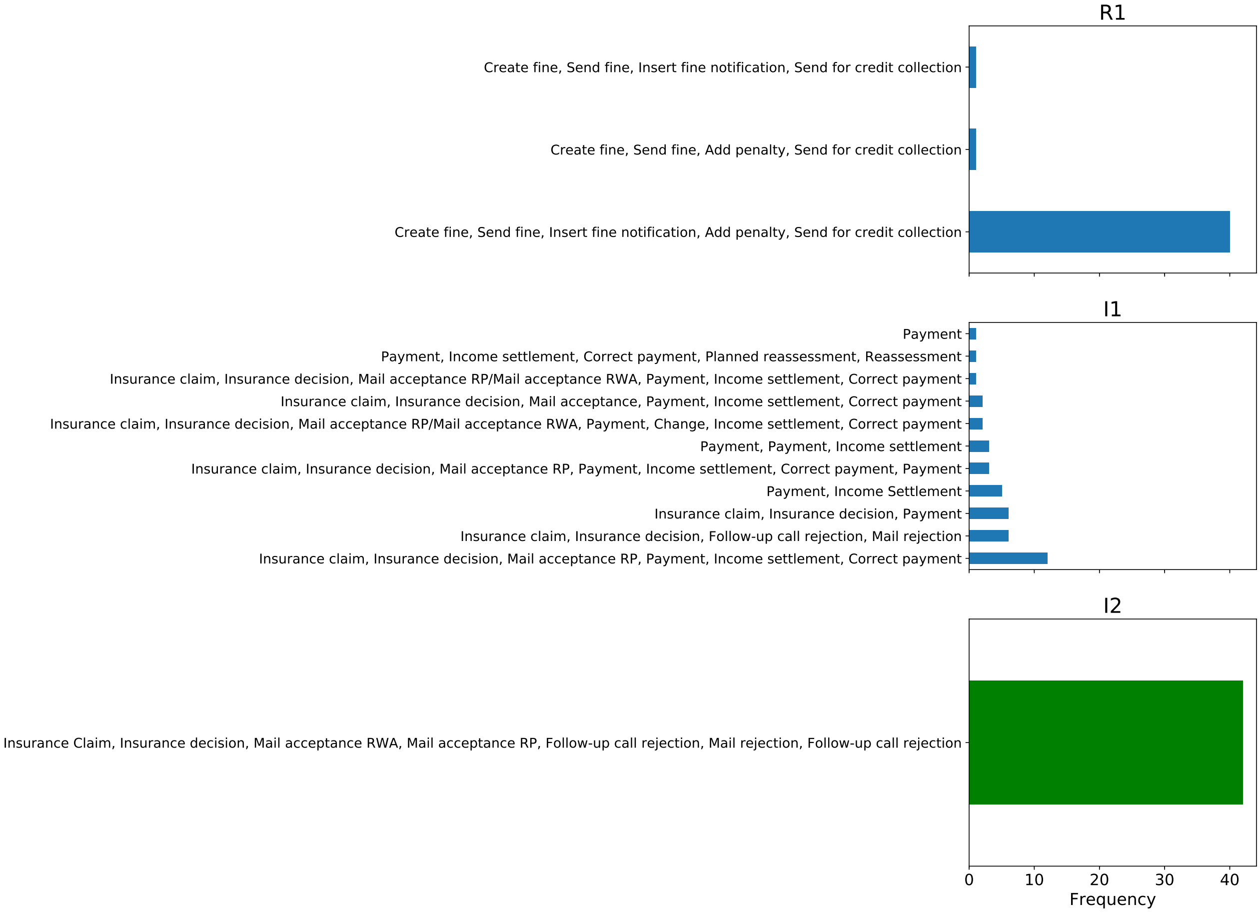


Figure 11: *An overview of the answers given to R1, I1, and I2. Green bars indicate correct answers while red bars indicate incorrect answers. Blue bars are neutral in the sense that there is no single correct answer for the respective question.*

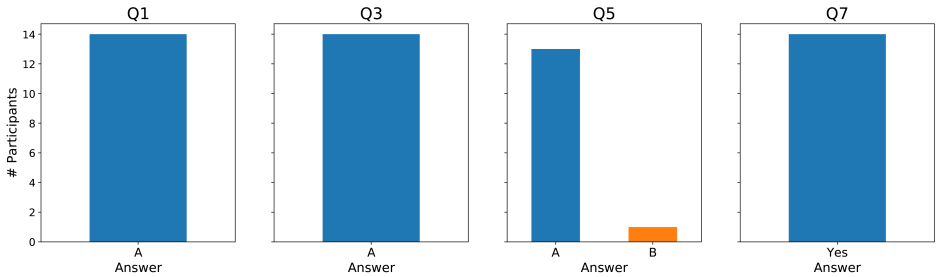


Figure 12: *An overview of the answers given to the (non-open) user study evaluation questions.*

5. Layout Examples

Presented below are figures in which two graph layouts of the same graph are shown. These graphs were obtained from some of the datasets as listed in Section 1. The graph layout on the left is always computed by *dot* while the graph layout on the right is computed by our algorithm.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

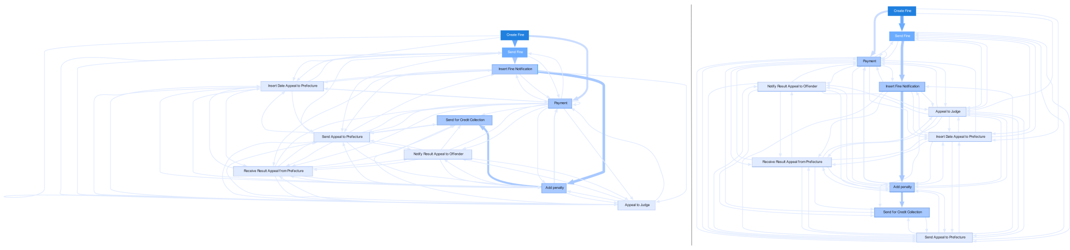


Figure 13: *Graph layouts for the complete Road dataset (which was used during the user study). In our layout (right), the 'main path' is in the center and aligned. In the layout computed by dot, this is not the case. Also, in our layout, the edges are more straight.*

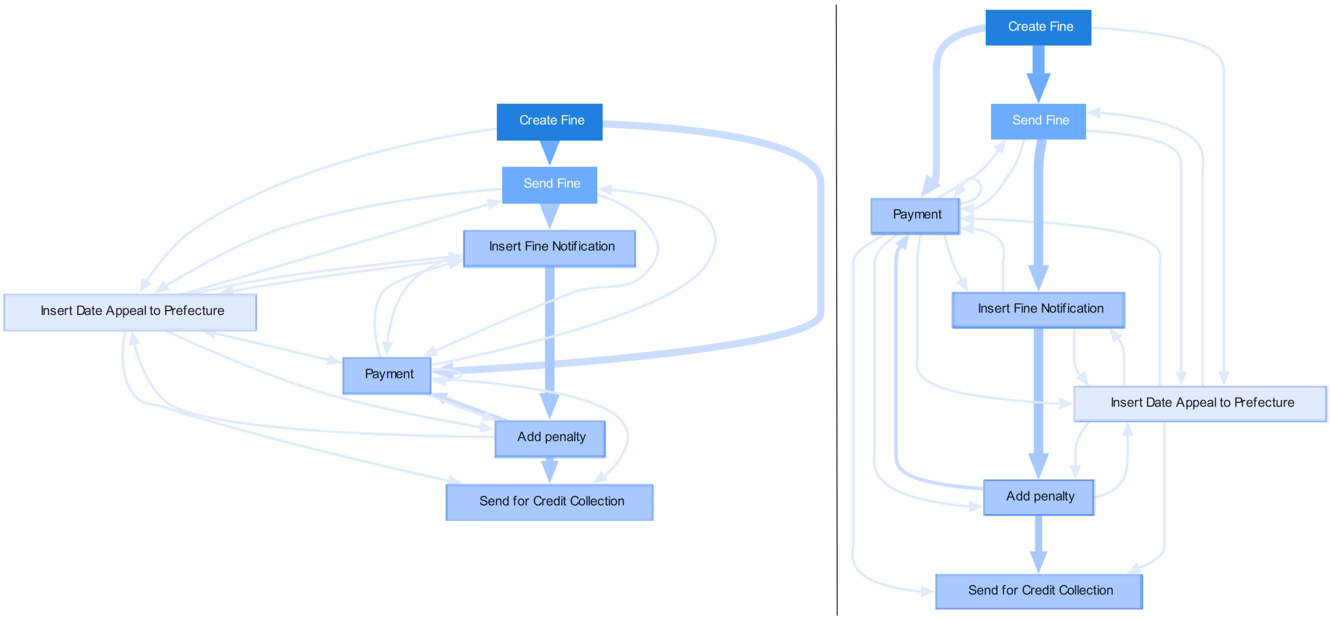


Figure 14: *Graph layouts for a subset of the Road dataset (which was used during the user study). In both layouts, the 'main path' is nicely aligned. In the layout computed by dot, however, the (salient) edge (Create fine, Payment) crosses the main path, which is quite distracting.*

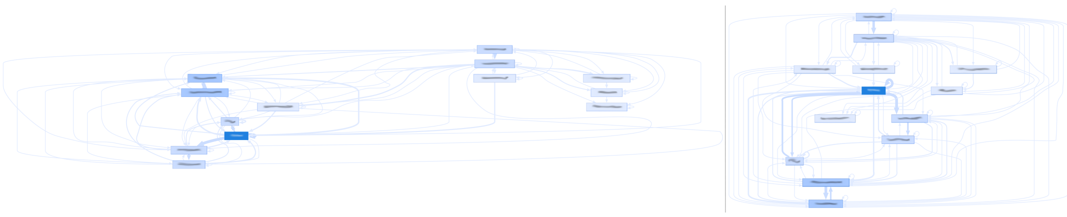


Figure 15: *Graph layouts for the complete Insurance dataset (which was used during the user study). Due to data privacy, node names have been blurred. In this dataset, there is no clear 'main path'. Nevertheless, in our graph layout, the edges are more straight. Also, the layout computed by dot is unnecessarily wide.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

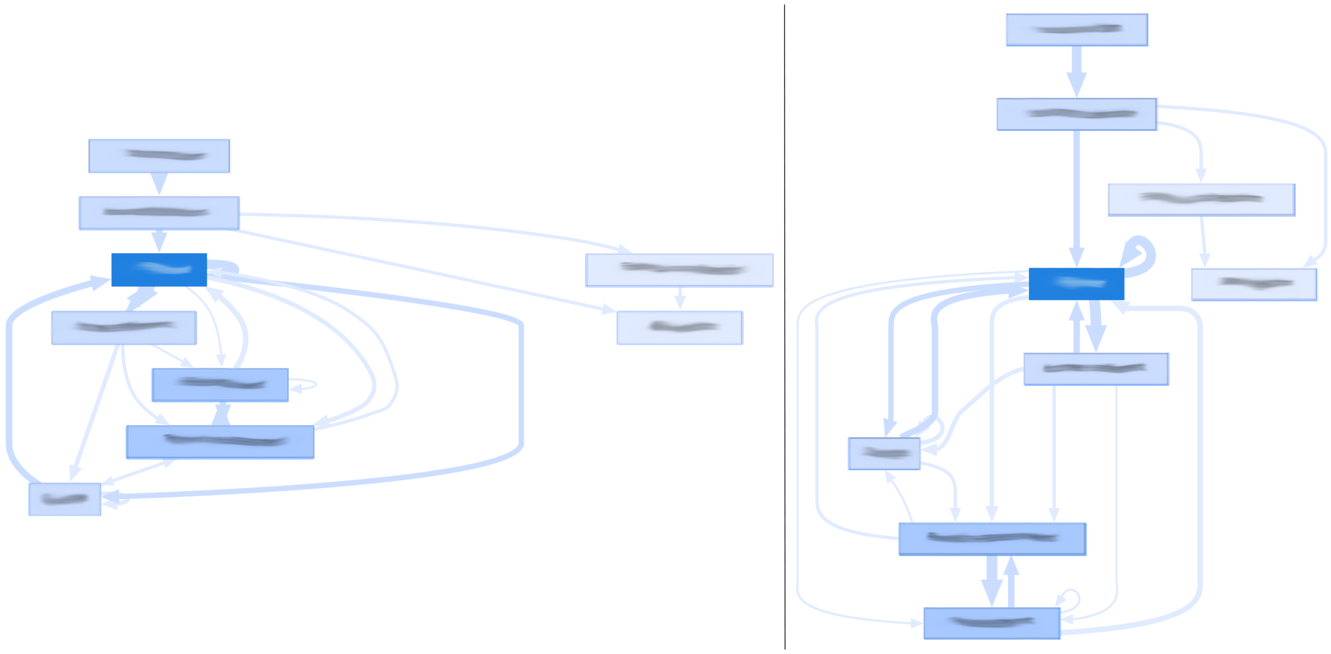


Figure 16: *Graph layouts for a subset of the Insurance dataset (which was used during the user study). Due to data privacy, node names have been blurred. Since this graph is not too large, both layouts are (quite) readable and understandable. In the layout computed by dot, however, there are some edges that overlap.*

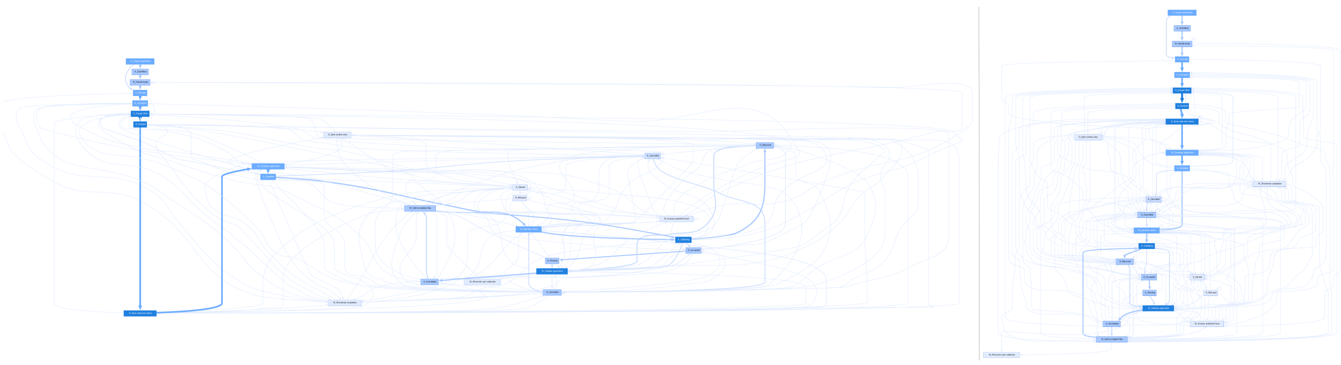


Figure 17: *Graph layouts for the complete BPI2017RCA dataset. In our layout, the 'main path' is nicely in the center and aligned. In the layout computed by dot, the beginning of the 'main path' is nicely aligned. After that, however, it is difficult to follow the main path.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

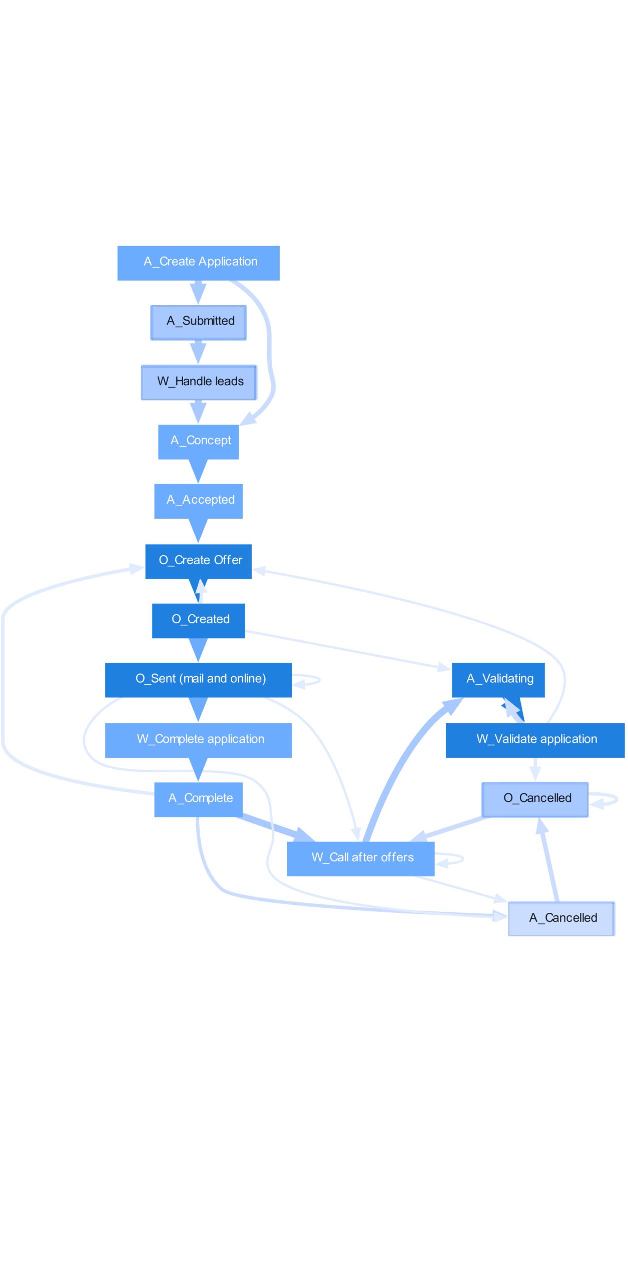


Figure 18: *Graph layouts for a subset of the BPI2017RCA dataset. In our layout, the 'main path' is nicely in the center and aligned. In the layout computed by dot, the beginning of the 'main path' is nicely aligned. After that, however, it is difficult to follow the main path.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

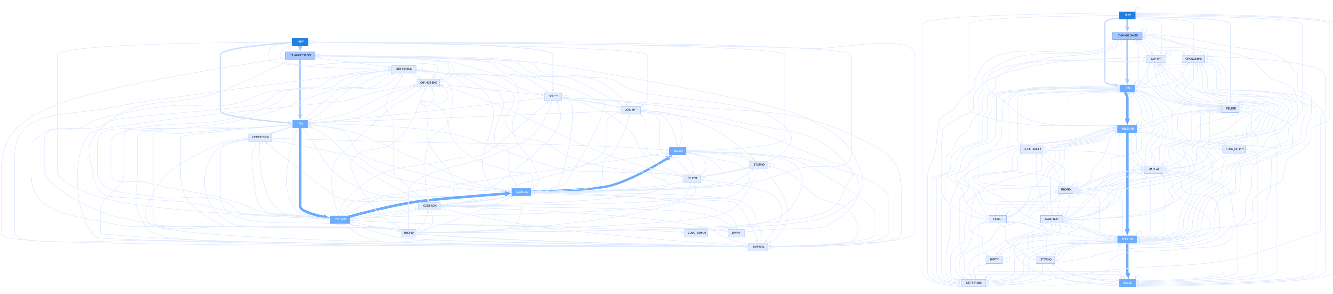


Figure 19: *Graph layouts for the complete Hospital dataset. In our layout, the 'main path' is nicely in the center and aligned. In the layout computed by dot, the 'main path' is not centralized and aligned.*

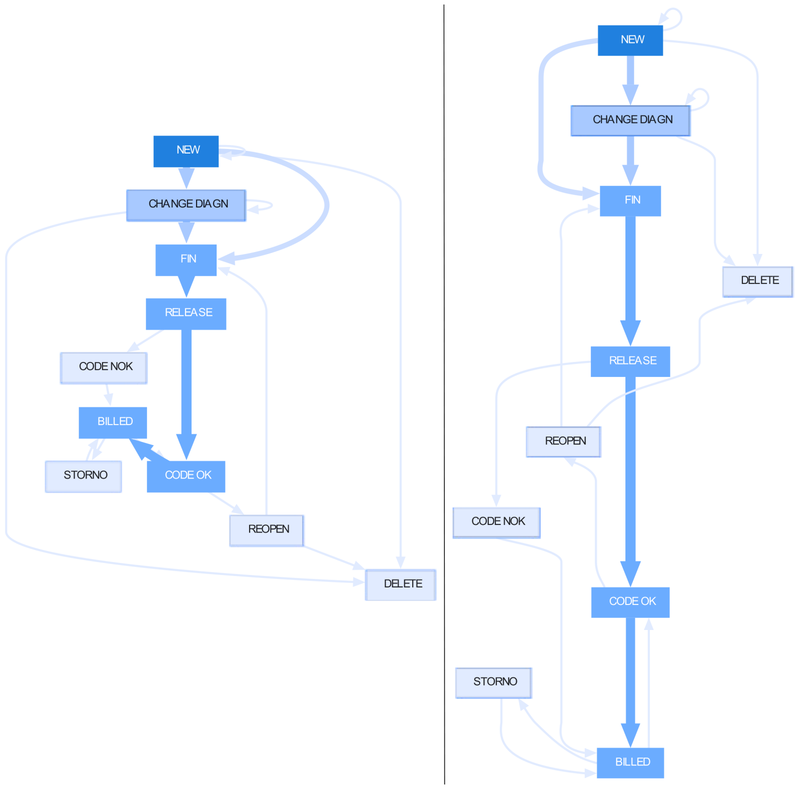


Figure 20: *Graph layouts for a subset of the Hospital dataset. In our layout, the 'main path' is nicely in the center and aligned (which is not the case in the layout computed by dot). The area of our layout, however, is quite large compared to the layout of dot.*

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

References

[3] BPI2012 - second international business process intelligence challenge. http://www.win.tue.nl/bpi/doku.php?id=2012:challenge. Accessed: 2018-05-12. 1

[4] BPI2013 - third international business process intelligence challenge. http://www.win.tue.nl/bpi/doku.php?id=2013:challenge. Accessed: 2018-05-12. 1

[5] BPI2014 - fourth international business process intelligence challenge. http://www.win.tue.nl/bpi/doku.php?id=2014:challenge. Accessed: 2018-05-12. 1

[6] BPI2017 - seventh international business process intelligence challenge. https://www.win.tue.nl/bpi/doku.php?id=2017:challenge. Accessed: 2018-05-12. 1

[7] BPI2018 - eighth international business process intelligence challenge. https://www.win.tue.nl/bpi/doku.php?id=2018:challenge. Accessed: 2018-05-12. 1

[AP16] ARCHAMBAULT D., PURCHASE H. C.: Can animation support the visualisation of dynamic graphs? *Information Sciences 330* (Feb. 2016), 495–509. 9

[BB99] BEDERSON B. B., BOLTMAN A.: Does animation help users build mental maps of spatial information? In *Proc. IEEE Symp. Information Visualization (InfoVis '99)* (1999), pp. 28–35. 9

[BPF14] BACH B., PIETRIGA E., FEKETE J. D.: GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (May 2014), 740–754. 9

[BS15] BERNSTEIN V., SOFFER P.: Identifying and quantifying visual layout features of business process models. In *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing. Springer, Cham, June 2015, pp. 200–213. 3

[BT98] BRIDGEMAN S., TAMASSIA R.: Difference Metrics for Interactive Orthogonal Graph Drawing Algorithms. In *Graph Drawing* (Aug. 1998), Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 57–71. 2, 4, 5

[BT00] BRIDGEMAN S., TAMASSIA R.: A User Study in Similarity Measures for Graph Drawing. In *Graph Drawing* (Sept. 2000), Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 19–30. 4, 5

[DG02] DIEHL S., GÖRG C.: Graphs, they are changing. In *Graph Drawing* (Aug. 2002), Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 23–31. 4, 5

[ELMS91] EADES P., LAI W., MISUE K., SUGIYAMA K.: *Preserving the mental map of a diagram*. Tech. rep., International Institute for Advanced Study of Social Information Science, 1991. 4

[GKNV93] GANSNER E. R., KOUTSOFIOS E., NORTH S. C., VO K. P.: A technique for drawing directed graphs. *IEEE Transactions on Software Engineering 19*, 3 (Mar. 1993), 214–230. 2

[Lev66] LEVENSHTEIN V. I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady 10* (Feb. 1966), 707. 5

[LMR98] LYONS K., MEIJER H., RAPPAPORT D.: Algorithms for Cluster Busting in Anchored Graph Drawing. *Journal of Graph Algorithms and Applications 2*, 1 (1998), 1–24. 4, 5

[MELS95] MISUE K., EADES P., LAI W., SUGIYAMA K.: Layout adjustment and the mental map. *Journal of Visual Languages & Computing 6*, 2 (June 1995), 183–210. 4

[PCJ97] PURCHASE H. C., COHEN R. F., JAMES M. I.: An Experimental Study of the Basis for Graph Drawing Algorithms. *J. Exp. Algorithmics 2* (Jan. 1997). 2

[Pur00] PURCHASE H. C.: Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with Computers 13*, 2 (Dec. 2000), 147–162. 2

[SIG07] SHANMUGASUNDARAM M., IRANI P., GUTWIN C.: Can smooth view transitions facilitate perceptual constancy in node-link diagrams? In *Proceedings of Graphics Interface 2007* (New York, NY, USA, 2007), GI '07, ACM, pp. 71–78. 9

[ZKS11] ZAMAN L., KALRA A., STUERZLINGER W.: The effect of animation, dual view, difference layers, and relative re-layout in hierarchical diagram differencing. In *Proceedings of Graphics Interface 2011 (GI'11)* (2011), Canadian Human-Computer Communications Society, pp. 183–190. 9

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.