VIPRA – EuroVis Workshop on Visual Process Analytics (2024)
A. Arleo, L. Pufahl, J. Rehse, F. Zerbato, S. van den Elzen, and T. von Landesberger (Organizers)

Visual Journey Analytics: lessons learned from real-world implementations

R. Brath[1], P. Andersen[1], M. Matusiak[1], and R. Gerber[2].

[1]Uncharted Software Inc., Canada
[2]Qualtrics, LLC., United States

Abstract

Process mining and more broadly journey analytics create sequences that can be understood with graph-oriented visual analytics. We have designed and implemented more than a dozen visual analytics on sequence data in production software over the last 20 years. We outline a variety of data challenges, user tasks, visualization layouts, node and edge representations, and interactions, including strengths and weaknesses and potential future research.

CCS Concepts
• ***Human-centered computing*** → ***Field studies;*** *Interaction design process and methods;* ***Visual analytics;***

1. Introduction

Visual journey analytics is the visualization and tightly coupled analytics associated with the sequence of steps taken through a process to achieve a goal. Journey analytics objectives include validation of journeys against expectations, deeper understanding of behavior, and data-driven decision making. Journey analytic applications occur in domains such as customer journeys through contact centers to increase sales or reduce churn; employee journeys for facilitating HR processes; patient journeys through a diagnosis, operation and recovery; or may occur in non-human processes, such as a courier shipment, a financial security settlement process, or an insurance claim process. Journey analytics can go beyond process analytics: rather than focus on a sequence of milestones in a defined process, journey analytics additionally capture behavioral characteristics of the user interactions and finer grain events between major milestones. The result of a process optimization may improve a process but not improve customer experience. Journey analytics seeks to improve the holistic journey of the user as well as the process.

The authors have extensive experience in developing visual analytics for journey analysis over the past two decades, with over a dozen applications deployed across various companies. Their contribution lies in offering a comprehensive overview of these implementations to identify common challenges in visual journey analytics.

2. Background

The origins of visual process analysis, in our opinion, reach back to milestones on Gantt charts [Cla35]; through a lineage of interactive event sequences such as LifeLines [PMR*96] or financial events [SB13]. In these cases, events are on a timeline, the notion of a specific sequence of events leading to a target outcome is not present.

More specifically, process mining extracts common sequences for users such as process discovery [ZKI19], conformance checking [RPGK22], system enhancement and so on. Research for mining of event sequences for process analysis is increasing. For example, 263 health-care process mining research papers reviewed by De Roock and Martin [DM22] started in 2005, with significant upward trend beginning in 2013. A few use visualization, for example, inductive visual miner [LFVDA14], commercial process mining software visualization by Celonis [ATP19], force-directed graphs [ZPP15], state-diagrams [AG18], and flowcharts [MYB*18].

There are many visualization techniques feasible, some of which depict the process sequence e.g. Sankey, chord diagram, force-directed graph, [SMNP18]; as well as supplementary data, e.g. via treemap, scatterplot, pie [GGJ*21, YM22]. Liu et al [LWD*16] use linear depictions of sequences with an overview of the most common high-level paths, and many specific sequences interactively aligned to an event of interest. To deal with high cardinality, they collapse frequently recurring subsequences (motifs), and sets of nodes that occur in sequence but vary in order (clusters). Furthermore, there exist commercial software for visualization of processes and journeys, including offerings from IBM, Microsoft, Celonis, Qualtrics, Medallia, CSG Systems, with some examples shown in Figure 1.

3. Review of Applications

We have faced challenges in aspects such as data, user tasks, and visualization across multiple real-world applications. We briefly

DOI: 10.2312/vipra.20241105

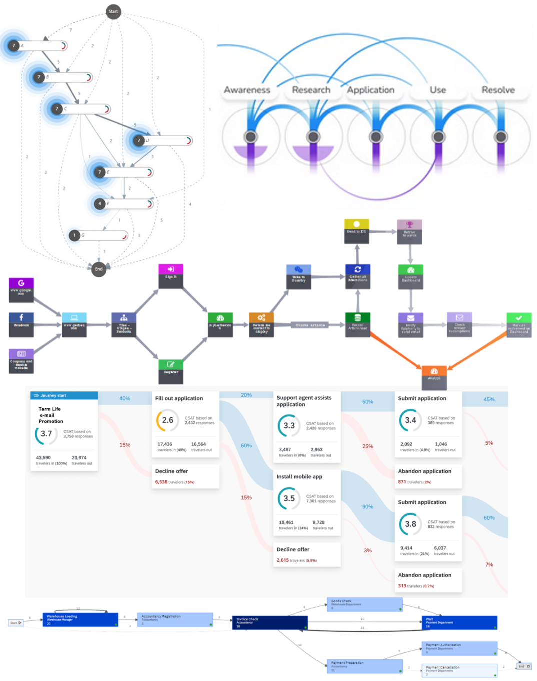


Figure 1: *Some examples of commercial process visualization from Microsoft, Medallia, Kitewheel, Qualtrics and IBM.*

touch on data and user task challenges before focusing on the specific issues with visual analytics.

3.1. Journey data challenges

One primary challenge in journey analytics is data discrepancy. Analysts anticipate a linear progression of events (e.g., A, B, C, D, E), however real-world data is far more complicated:

- *Joining data*: In journey analytics, a key challenge is joining data from many systems. Along with process data, customer journey analytics incorporates behavioral data from various channels (e.g., web, mobile, social), each with its own schema. Aligning customer identifiers can be problematic, e.g. across channels or when customers don't log in on some platforms.
- *Temporal frequency*: Data from different systems have differences in delays: some may be near real-time, others batch update hourly or daily. This makes it difficult to use journey analytics for monitoring, real-decisioning, and throttles analytics to the slowest data to have appropriately sequenced data.
- *Data not organized as steps*: For example, data from web traffic is individual pages. Other than the *shopping cart* process, the larger journey through steps such as discovery, exploration, consideration, may be difficult to categorize. This requires an extensive effort to either categorize the content upfront so the appropriate category is logged, or extensive effort to post-process the granular data into the higher-level steps.
- *Data does not follow sequence*: It is not uncommon to have duplicate steps, reversals, skips, out-of-order steps and so forth (e.g. AAABCDE, ABCDCBCDE, ADBCE, ACE). Customers may reconsider, backtrack and modify a purchase; a patient may be re-diagnosed; different systems may process events at different time horizons; and so on. In one application, we abandoned the notion of an *ideal path*: fewer than 1% of actual journeys followed the analyst's idealized path.
- *Additional step types*: Despite the notion of a set of steps, there inevitably are anomalous steps which may occur in a sequence (e.g. ABCXE). A container does not move directly from port to port but goes through an intermediate port transferring from ship to ship; or a customer does a login; or a patient has a procedure which does not normally occur in the expected process.
- *Unstructured data:* How customers, patients, employees feel about a journey is important to perception of success. Unstructured data such as surveys provides insights that cannot be uncovered with a purely metrics oriented approach.

A further challenge is to manage this data at scale, e.g. across millions of journeys. With current hardware, this scale of data cannot fit within client-side browser memory, and instead requires server-side interactive processing of the paths. The result is an incredibly wide variety of permutations of unique paths.

3.2. Journey analytic tasks

There are many tasks with journey data beyond idealized path and event confirmation. Key "why" tasks include [BM13]:

- *Process discovery*: In some cases, analysts have only hypotheses regarding the process sequence and want visualization to comprehend the system.
- *Comparison*: One may need to see multiple states: e.g. actual paths vs mental model (conformance checking [RPGK22]); before and after process modification; evaluation of alternatives such as a simulation; journey differences in sub-populations, etc.
- *Decomposition*: One may want a high level journey across the full customer lifecycle, and successively decompose that to lower-level journeys, such as customer service journey, and further to a payment dispute journey, etc.
- *Problems and intervention*: Journey visualization can reveal friction, dormancy, journey switches, or other issues. Visual representations aid analysis into root causes of problems with aggregate views. A journey visualization provides a convenient point of access to intervene in a journey: to modify journeys in progress, or to modify the process.
- *Operations and risk*: Many of these processes can be monitored in real-time, trigger alerts and indicate areas of emergent risk.
- *Predictive simulation and optimization*: The analytic model can aid prediction: e.g. forecasting how many email invites will reach the shopping cart next week; estimating the time before emergency services are overwhelmed during a pandemic. Optimization may be used to adjust offerings: e.g. determining the mix of offers to send to acquire 1000 mortgages and 1000 loans.

3.3. Visual representation of layout

Given the variety of tasks and potential variation in data, we've created a wide variety of different visualizations of the process sequence steps, as indicated in Figure 2.

We use diagrams in this paper as most implementations are

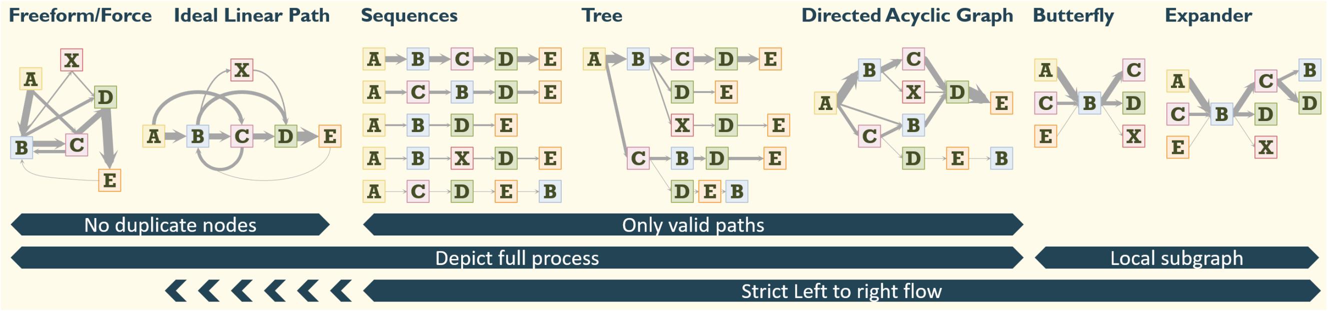


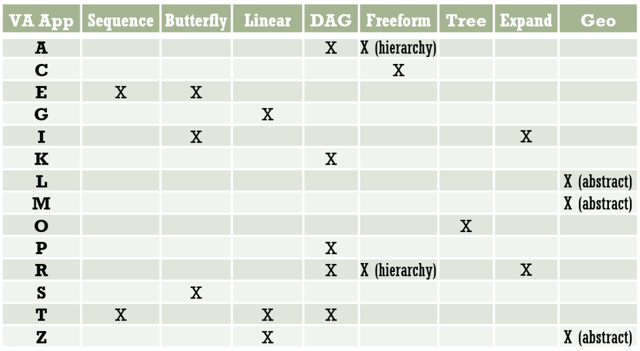
Figure 2: *Diagrams of visualization variants. All indicate the same toy dataset, shown under Sequences, with color-coded process steps and edge-thickness indicating flow volume. Span annotations indicate properties of representations.*

confidential (or previously published [KGVW21, HRGK*21, BJ15, JLGW14]). All diagrams show data from a process with six unique steps (A-X), five unique sequences, and flow volume indicated by edge thickness. The annotations at the bottom of the diagram indicate common properties:

- *No duplicate nodes*: The first two representations do not duplicate nodes. This aids visually assessing how one arrives and departs from a state as well as summary properties per node. The unique paths are not recoverable from this representation.
- *Only valid paths*: The middle three representations explicitly depict only valid sequences. These are useful when one wants to understand the paths, although duplication of nodes within the diagram increases cognitive load during path comparison tasks.
- *Local subgraph*: The right two representations depict only a portion of the paths; around a node or subpath of interest. The context of the larger graph and actual paths are missing.
- *Full graph*: The first five representations show the full graph. The overview is useful to understand the full process.
- *Left to right flow*: Except for the first freeform representation, most representations attempt to create a left-to-right flow. It is desirable to have the sequence flow in one direction to align with the mental model of the users.

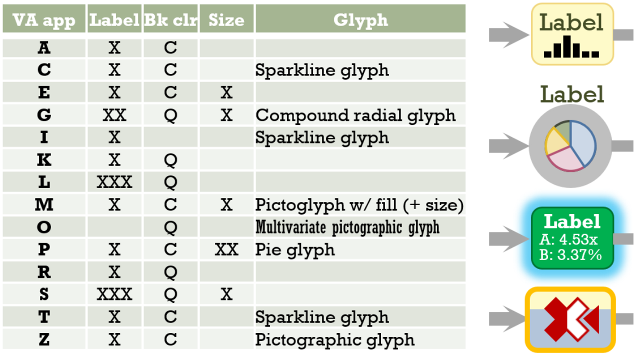
These common properties indicate desirable criteria for sequence visualization. Not all can be achieved in one visualization, and each representation has pros and cons, as seen in practice.

- *Freeform*: The freefrom representation is familiar to systems modelers as it is highly similar to *state transition diagrams*. It can be useful for complex systems with feedback loops such as systems dynamics and biologic processes. These diagrams risk becoming confusing spaghetti, e.g. force-directed layout algorithms can often result in difficult to comprehend graphs. We note that human-curated complex graphs can be very high quality (e.g. [MS12]), but require extensive effort to create.
- *Ideal path*: The ideal path is explicitly depicted as a left to right sequence (or top to bottom). Other edges hop, loop or go backwards. It is similar to *syntax diagrams*, *railway diagrams* and *process flowcharts*. This is the only representation that makes the expected flow visually dominant: the Gestalt effect of continuation is very strong. This representation is liked. Skips, backflows and deviations can become disorienting, but interactions can hide these if desired. Furthermore, unique paths are not visible: the path ABCBCBCBCDE can be constructed but may not exist in the source data. If there are few backflows and alternative steps, the ideal path may resemble a Sankey diagram.
- *Sequences*: is a raw depiction of unique event sequences. Explicitly showing each path is useful for detailed inspection without aggregation, e.g. examining path properties under one specific sequence. This method does not scale well to thousands of unique paths, and visually becomes difficult to compare duplicate nodes, particularly when there are more than 10 or so unique node types.
- *Tree*: collapses common nodes at the start of the path. Multiple trees may exist, as processes do not always start at the same node. Trees often become wide, although a radial layout can help.
- *Directed Acyclic Graph* (DAG): collapses common segments across the graph. Multiple valid DAGs may exist. With many paths, multiple disconnected DAGs are common. Note nodes and edges may appear multiple times, e.g. node B occurs three times, edge AB occurs twice. The DAG can be represented as a Sankey diagram. This method can be adapted so that each node only occurs once, but introduces backward flows, causing loops and losing strict "only valid paths" criteria. This works with few backwards flows, but can quickly turn into spaghetti diagrams if many backflows occur.
- *Butterfly* focuses on a singular node and typically expands one tier in both directions. It is typically very easy to understand, although some users are unaware that non-existent paths are represented (e.g. EBC does not exist). Typically picking any visible node promotes that node to the center and redraws the butterfly around the new node. As it is very local (i.e. the neighborhood around a single node), users may need to traverse around the local graph back and forth to gain a broader understanding.
- *Expander* is an interactive extension of the butterfly. A click expands one node further. This can be repeated further. Constraints may be applied such that the primary path only contains valid paths and invalid branches pruned.
- *Geographic*: is not shown in the diagram, but some processes have physical real-world locations, such as ports or hospitals. These processes can be overlaid on the physical coordinate space - i.e. depicting the process on a map. Note that maps may be abstracted and the coordinate space adjusted, e.g. to reduce whitespace or regularize the layout.
- *Adjacency matrix*: was not used in any of our applications.



VA App	Sequence	Butterfly	Linear	DAG	Freeform	Tree	Expand	Geo
A				X	X (hierarchy)			
C					X			
E	X	X						
G			X					
I		X					X	
K				X				
L								X (abstract)
M								X (abstract)
O						X		
P				X				
R				X	X (hierarchy)		X	
S		X						
T	X		X	X				
Z			X					X (abstract)

Figure 3: *Visualization layouts for 14 applications.*



VA app	Label	Bk clr	Size	Glyph
A	X	C		
C	X	C		Sparkline glyph
E	X	C	X	
G	XX	Q	X	Compound radial glyph
I	X			Sparkline glyph
K	X	Q		
L	XXX	Q		
M	X	C	X	Pictoglyph w/ fill (+ size)
O		Q		Multivariate pictographic glyph
P	X	C	XX	Pie glyph
R	X	Q		
S	XXX	Q	X	
T	X	C		Sparkline glyph
Z	X	C		Pictographic glyph

Label
Label
Label
A: 4.53x
B: 3.37%

Figure 4: *Node representations per application including label, color, size and glyph (if any); with examples on right.*

The use of these layouts in practice in 14 visual analytic applications is summarized in Figure 3.

3.4. Visual representation of nodes and edges

The previous section discussed the overall layout and used a trivial representation of a box with a letter for nodes. In practice, node representations can be complex containing a variety of categoric, quantitative and textual data, as shown for each of our visualizations in figure 4. The right side shows four diagrammatic node representations. Some observations:

- *Labels*: are used in 13/14 applications. Multiple applications use more than one label per node indicating additional information such as metrics or conditions in three applications (as shown in the green node with the blue glow). Only one application does not use labels: however it has a complex glyph with attributes such as outline color, partial fill in the glyph background, and a compound symbol with partial fill.
- *Color*: in 13/14 applications denotes data, evenly split between use of color to encode categoric (C) vs. quantitative (Q) data.
- *Size*: is a potent visual cue, but can be disruptive to layout. It works well in the butterfly layout, or constrained to a limited range. The second example node, a pie chart, uses size twice: setting overall node size based on total event count; and a smaller inset for a filtered sub-population.
- *Glyphs*: are used to represent multivariate data per node. There is an incredible variety of the glyphs across applications as might be expected [BKC*13]. Glyphs can show timeseries associated with each node, thus depicting graph properties that vary over time which can be otherwise difficult to depict, e.g. via animation or small multiples [Jon12]. Glyphs may include various indicators, such as: badges; one or more icons/pictographs with variants such as separate outline, fill and partial fill; a background with separate fill and outline; and so on. Multivariate icons can be challenging to design so that they can easily be decoded [Bra15].
- *Other visual attributes*: Glow was used in only one application to denote an alert. Two applications used a 3D layout with 3D glyphs; although in one of those applications the 3D was removed when the application was deployed.

Edges in most applications tend to use simple representations of flows between nodes. Attributes used for edges include:

- *Thickness*: is used in 11/14 applications to indicate quantity.
- *Color*: is used to indicate either categoric data or quantitative data (e.g. length of time between transitions).
- *Length*: in one application, edge length was used to indicate the length of time between transitions (note the variation in edge length in the Tree diagram).
- *Dash*: in two applications we used dash patterns to indicate data.

Note that the above diagrams show a trivial dataset and simplified layout. In practice, the visualizations can be complex: with more nodes and edges, additional elements in the layout, and a greater variety of glyphs, such as the example in figure 5.

3.5. Interactions

Interactions are key to making a representation usable for the tasks. Given the scale of data, there may be significant effort to achieve interactive response times. Figure 6 shows an application with rich interaction on datasets with billions of transactions.

- *Filter/Collapse/Hide/Etc.*: Given the variety of permutations many of which may not be of interest, UX to control what is desirable to be seen or not is required. For example, some tasks

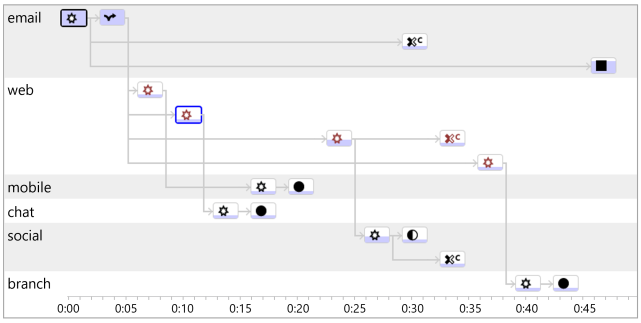
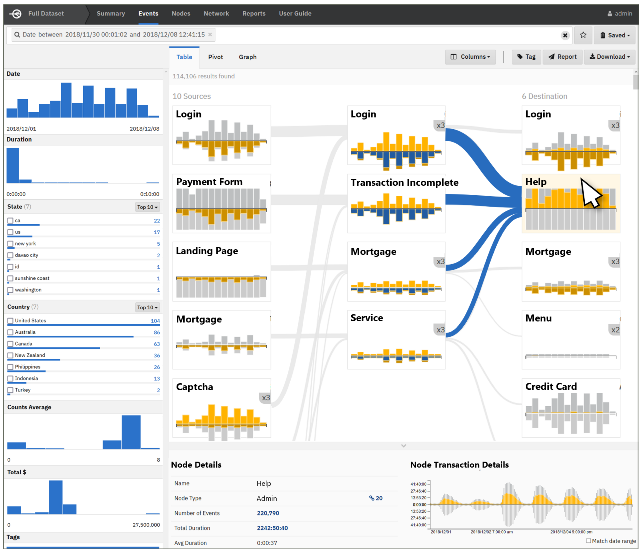


Figure 5: *A tree layout, organized into swim-lanes and a horizontal time axis; with multivariate glyphs per node.*



Date between 2018/11/30 00:01:02 and 2018/12/08 12:41:15

Node Details

Name	Help	
Node Type	Admin	20
Number of Events	220,790	
Total Duration	2242:50:40	
Avg Duration	0:00:37	

Figure 6: *A full application, with butterfly-variant process visualization, with search, filter, path profiling (left panel), selection of subpaths (highlight) and drilldown (bottom panel).*

need to assess repeat process steps (AAA) or backwards process steps (ABCA), while some tasks never consider repeats or backwards. The starting-point and/or ending-point may be very important to define, e.g. for a given treatment, start with the first diagnosis, end with the third prescription.

- *Paths*: can be overlaid to indicate the actual or predicted paths forecast; either globally (top overall path) or locally (predicted paths from a given point forwards).
- *Comparison*: In a number of cases, the visualization needs to aid comparison of different states, e.g. before and after a modification in the system; change in system behavior between this year and last year; and so on.

4. Discussion

At least a half dozen of these applications have been deployed for more than half a decade; one has over 1000 users. Enduring applications include simpler representations like butterfly, expander, DAG, and linear. These representations have strong visceral appeal and are easily understood with minimal explanation. Furthermore, insight can be derived without needing many clicks. Successful representations match user tasks and mental models closely, answering questions with minimal interaction.

We have seen peer applications (e.g. Figure 1) use similar data to do similar analyses. Sometimes these peer applications are close copies (same layout, similar glyph); sometimes they use a different layout. For example, in one case we used a DAG, and a peer application used an expander layout. In another application we used a butterfly very successfully, and a peer application instead used a tree. This suggests that there may be potential for a singular visualization approach across uses (or at a smaller set). Regardless - given each project has significantly different characteristics (e.g.

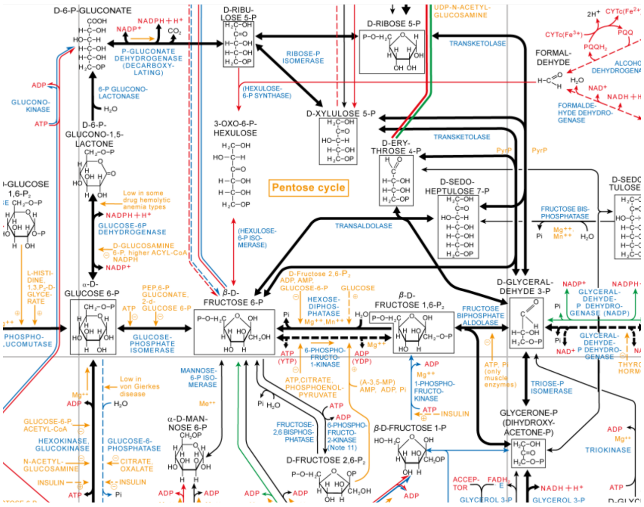


Figure 7: *Human curated metabolic pathway diagram scales to highly complex processes.*

number of unique steps, number of permutations, range of data per edge and node, temporal data, breadth of potential analytic tasks) - each project needs to move through an initial requirements phase to understand the goals, data, tasks, and so on, and this tends to lead to a unique solution.

In all cases, these representations have challenges scaling to processes with many steps: 40 or more. Permutations are enormous making it difficult for any full layout to clearly depict these. However, human curated metabolic pathways diagrams in biology depict complex non-linear processes with thousands of nodes clearly, e.g. a small portion is shown in Figure 7 [MS12]. These diagrams use techniques such as edge routing to minimize crosses and occlusion, map-like overview and zoom to details, color-coding and line styles to aid tracing paths, labels packed densely without contraction, and so on.

5. Conclusions

In addition to the present challenges of operating at scale, future research should include developing scalable solutions that cater to diverse domains. The exploration of a reusable visualization framework holds promise in facilitating cross-domain applicability, enabling insight and analyses across disciplinary boundaries. Our experience with visual journey analytics has indicated strong interest by analysts and such approaches would not only enhance the efficiency of visualization tools but also foster possible collaboration and knowledge exchange among disparate fields.

Furthermore, there are a variety of uses we have not yet considered. Our current focus has primarily involved desktop applications; however integrating mobile requirements could potentially unlock new avenues for real-time data analysis and decision-making, thus enriching the utility and effectiveness of visualization tools in diverse operational settings.

References

[AG18] Arnolds I. V., Gartner D.: Improving hospital layout planning through clinical pathway mining. *Annals of Operations Research 263* (2018). doi:10.1007/s10479-017-2485-4. 1

[ATP19] Aguirre J. A., Torres A. C., Pescoran M. E.: Evaluation of operational process variables in healthcare using process mining and data visualization techniques. *Health 7* (2019), 19. 1

[BJ15] Brath R., Jonker D.: *Graph analysis and visualization: discovering business opportunity in linked data*. John Wiley & Sons, 2015. 3

[BKC*13] Borgo R., Kehrer J., Chung D. H., Maguire E., Laramee R. S., Hauser H., Ward M., Chen M.: Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics State of the Art Reports* (May 2013), EG STARs, Eurographics Association, pp. 39–63. URL: http://diglib.eg.org/EG/DL/conf/EG2013/stars/039-063.pdf. 4

[BM13] Brehmer M., Munzner T.: A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics 19*, 12 (2013), 2376–2385. 2

[Bra15] Brath R.: High category glyphs in industry. *Visualization in Practice at IEEE VisWeek* (2015). 4

[Cla35] Clark W.: *The Gantt Chart: A Working Tool of Management*. Sir Isaac Pitman & Sons, 1935. 1

[DM22] De Roock E., Martin N.: Process mining in healthcare – an updated perspective on the state of the art. *Journal of Biomedical Informatics 127* (2022), 103995. URL: https://www.sciencedirect.com/science/article/pii/S1532046422000119, doi:https://doi.org/10.1016/j.jbi.2022.103995. 1

[GGJ*21] Guo Y., Guo S., Jin Z., Kaul S., Gotz D., Cao N.: Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2021), 5091–5112. 1

[HRGK*21] Husain F., Romero-Gómez R., Kuang E., Segura D., Carolli A., Liu L. C., Cheung M., Paris Y.: A multi-scale visual analytics approach for exploring biomedical knowledge. In *2021 IEEE Workshop on Visual Analytics in Healthcare (VAHC)* (2021), IEEE, pp. 30–35. 3

[JLGW14] Jonker D., Langevin S., Gauldie D., Wright W.: Influent: Scalable transactional flow analysis with entity-relationship graphs. *Poster Proc. EuroVis* (2014). 3

[Jon12] Jonker D.: Linked visible behaviors: A system for exploring causal influence. AHFE Cross-Cultural-Decision-Making (CCDM) conference, 2012. 4

[KGVW21] Kapler T., Gray D. W., Vasquez H., Wright W.: Causeworks: A framework for transforming user hypotheses into a computational causal model. In *VISIGRAPP (3: IVAPP)* (2021), pp. 50–63. 3

[LFVDA14] Leemans S. J., Fahland D., Van Der Aalst W. M.: Process and deviation exploration with inductive visual miner. In *12th International Conference on Business Process Management, BPM 2014* (2014), CEUR-WS. org, pp. 46–50. 1

[LWD*16] Liu Z., Wang Y., Dontcheva M., Hoffman M., Walker S., Wilson A.: Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE transactions on visualization and computer graphics 23*, 1 (2016), 321–330. 1

[MS12] Michal G., Schomburg D.: *Biochemical pathways: an atlas of biochemistry and molecular biology*. John Wiley & Sons, 2012. URL: https://biochemical-pathways.com/#/map/1. 3, 5

[MYB*18] Metsker O., Yakovlev A., Bolgova E., Vasin A., Koval-chuk S.: Identification of pathophysiological subclinical variances during complex treatment process of cardiovascular patients. *Procedia computer science 138* (2018), 161–168. 1

[PMR*96] Plaisant C., Milash B., Rose A., Widoff S., Shneiderman B.: Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1996), pp. 221–227. 1

[RPGK22] Rehse J.-R., Pufahl L., Grohs M., Klein L.-M.: Process mining meets visual analytics: the case of conformance checking. *arXiv preprint arXiv:2209.09712* (2022). 1, 2

[SB13] Sorenson E., Brath R.: Financial visualization case study: Correlating financial timeseries and discrete events to support investment decisions. In *Information Visualisation (IV), 2013 17th International Conference* (2013), IEEE, pp. 232–238. 1

[SMNP18] Sirgmets M., Milani F., Nolte A., Pungas T.: Designing process diagrams–a framework for making design choices when visualizing process mining outputs. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I* (2018), Springer, pp. 463–480. 1

[YM22] Yeshchenko A., Mendling J.: A survey of approaches for event sequence analysis and visualization using the esevis framework. *arXiv preprint arXiv:2202.07941* (2022). 1

[ZKI19] Zayoud M., Kotb Y., Ionescu S.: β algorithm: A new probabilistic process learning approach for big data in healthcare. *IEEE Access 7* (2019), 78842–78869. doi:10.1109/ACCESS.2019.2922635. 1

[ZPP15] Zhang Y., Padman R., Patel N.: Paving the cowpath: Learning and visualizing clinical pathways from electronic health record data. *Journal of biomedical informatics 58* (2015), 186–197. 1