A

1. Overview

The authors presented the design and implementation of a comprehensive visualization system - ViDX. ViDX is used to perform real-time performance monitoring of the production line, as well as analyzing historical data to diagnose inefficiencies, locate anomalies and form hypothesis of their root-causes [1].

This is a problem-driven work using top-down design approach. The authors engaged with managers and operators at the production line to understand the line issues, and then set out the design requirements for the visual solution. The design key components are historical data analysis using Marey's Graph and real-time assembly line performance tracking using Radial Graph. A web application is designed to access the visualization.

2. Domain Situation

The domain of the application is big data analytics in Industry 4.0. The paper explains clearly the needs of performance monitoring for assembly line in smart factories, where all machines and sensors are interconnected, and are producing large amount of data every single second. Key concerns are overall equipment efficiency (OEE) and temporal productive maintenance (TPM), which directly affects the cost and profit of the company.

The paper shares some successful use cases of statistical methods that track production processes and analyze factors related to the yield in pharmaceutical and mining industries. The paper points out that the closest works [2][3] in the same application domain still remain at planning and simulation stages in manufacturing. However, both works are fundamentally different from this research because the authors actually design a visual analytic solution for modern smart factories using the data collected during the operation of the production line. The paper also discusses temporal data visualization which is one critical component of the design. This is because operations and processes in an assembly line are essentially sequences of pre-scheduled events, ideally taking place in predictable and timely manners.

The common threat is that the problem is mischaracterized where the target users do no in fact have these problems. The designers have done a very good job in this aspect. The designers worked very closely with the managers and operators of a Bosch manufacturing plant throughout the six months project period to gather requirements. During the project period, designers were allowed to access the end users' production databases. Designers also had frequent (approximately bi-weekly) video conferences and in-person meetings as well as email discussions. The meetings usually involved a manager responsible for the "Big Data in Industry 4.0" program in the company and technical staff responsible for the design and maintenance of the databases. By doing this, the designers were able to understand precisely the problems and needs of the users at this application domain, as summarized below:

1. Historical data analysis
   a. Detect inefficiencies and anomalous events.
   b. Engage users to detect outliers interactively.
   c. Support predictive analysis.
2. Real-time tracking of assembly line performance
   a. Highlight anomalies in real time.
   b. Visually indicate problematic components in 3-D models.
   c. Support smooth exploration and interaction of big data.

3. Task and Data Abstraction

In this paper, the focus is to design an informative and intuitive visualization interface for both real-time monitoring of assembly line performance and historical data analysis. The data to be collected is the cycle time. All manufacturing process data are stored in a relational database. The data is indexed by timestamp to support efficient retrieval of data within a specified time interval. The temporal data is indeed a time-series dataset, namely, an ordered sequence of time-value pairs. These dataset are a special case of tables, where time is the key. Datasets are also dynamic stream where humungous of data are generated every single second in the assembly line.

The authors further abstracts assembly lines into directed acyclic graphs (DAGs). The nodes in the DAGs are the workstations. The edge in the graph indicates the ordering of workstations. This is illustrated in Figure 3, where Sx refers to a workstation.

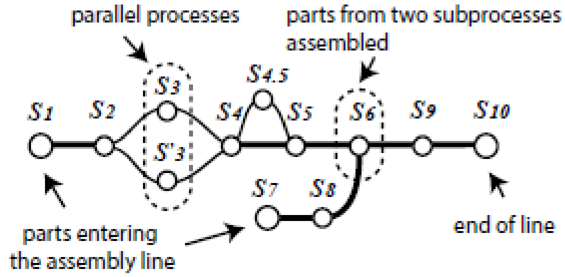

Figure 3. A directed acyclic graph (DAG) view of an assembly line. From [Xu et al., Figure 1].

Programmable logic controllers (PLCs) are widely used to automate the machineries in each station. PLCs record when each part is moved onto a station. Based on time collected, designers can calculate the cycle time of the part on a particular station. From the cycle time, the worker can identify outliers and abnormal processes. PLCs also record fault codes if any error occurs when a part is processed in a particular station. The timestamps and fault codes are used as the trace data of the corresponding part.
In order to fulfil the requirements from the users, the following task abstractions are identified:

User Requirements	Actions	Targets
Historical data analysis		
Detect inefficiencies and anomalous events	Analyze: Consume-Discover Search: Lookup, Browse, Locate, Explore Query: Identify, Compare, Summarize	All Data: Trends, Outliers Attributes: Distribution, Extremes
Engage users to detect outliers interactively	Produce: Annotate, Record	All Data: Outliers
Support predictive analysis	Analyze: Consume-Discover Search: Lookup, Browse, Locate, Explore Query: Identify, Compare, Summarize	All Data: Trends, Outliers, Features Attributes: Distribution, Extremes, Dependency, Correlation, Similarity
Real-time tracking of assembly line performance		
Highlight anomalies in real time	Analyze: Consume-Discover, Consume-Enjoy	All Data: Trends, Outliers
Visually indicate problematic components in 3-D models	Search: Locate	-
Support smooth exploration and interaction of big data	-	-

Table 1: Task Abstraction

Since this is a huge system design, more than just visualization aspects are considered. Some requirements such as smooth exploration support, need good infrastructure and network designs.

At the abstraction level, the threat is that the identified task abstraction blocks and designed data abstraction blocks do not solve the characterized problems of the target audience. The key aspect of validation against this threat is that the system must be tested by target users. The authors particularly shine in this aspect. First of all, immediate validation approach is used at the beginning of the project stage. The technical staffs responsible for the design and maintenance of the databases are key participants in the technical meeting during the project period. The discussions in the meetings are mostly about the semantics of the data attributes when the authors started building the system, and more about the feedback of the prototype at later stages.

A downstream validation is also performed at the end of the design, where two detailed assessments were performed to evaluate the design:

1. Run real case studies that illustrates assembly line performance issues.
2. Conduct a pilot run in the manufacturing line.

The authors also conducted in-depth interviews with a large group of end users after each assessment. This is to validate the design decisions and assess the effectiveness of the system. Overall, the feedback is encouraging.

4. Visual Encoding and Interaction Idiom

Marey's graph, which is traditionally used for depicting bus or train stops scheduling, is applied to the assembly line, where each station is viewed as a bus or train stop. By doing this, the polylines in the graph will show the complete history of a product in the assembly line. This allows reveal of out-of-order events such as delays and halt of production line. This visual encoding method is good because the end users know exactly what has happened by just looking at the graph without any calculations.

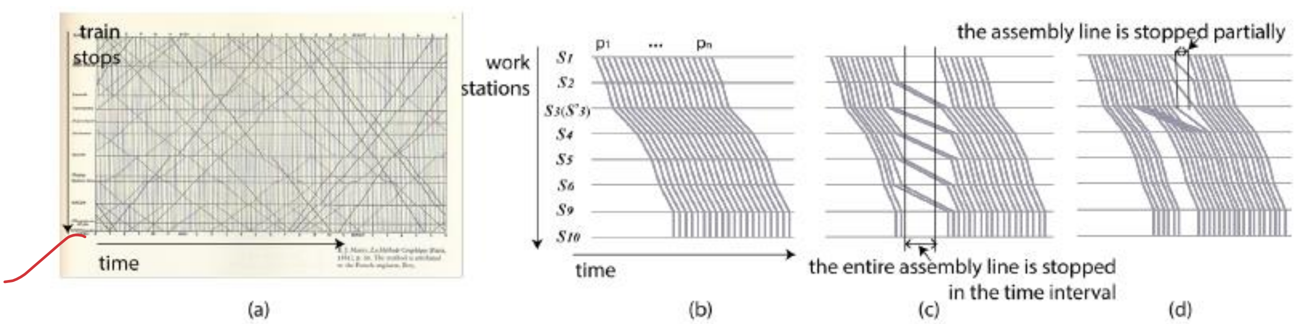

Figure 4. Marey's graph and visual patterns when a product starts being processed on each workstation on an assembly line. It uses the DAG in Fig. 3 for illustration. (a) Original Marey's graph; (b) Visual pattern showing smooth assembly line; (c) The assembly line is completely stopped during a time interval; (d) The assembly line is partially stopped to handle unprocessed products. From [Xu et al., Figure 3].

To avoid visual clutter which may affect the visibility of outliers, the authors introduce a time-aware outlier-preserving visual aggregation technique to enhance it. Basically polylines of adjacent normal processes are aggregated and shown as a thick band, while preserving each outlier as an individual polyline in the graph. This enhancement is critical because there could be thousands or millions of normal polylines in the assembly line, so without doing anything on this would definitely drive users away due to poor visibilities.

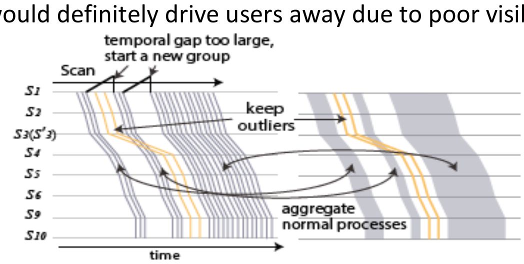

Figure 5. Time-aware outlier-preserving visual aggregation technique. From [Xu et al., Figure 3].

The authors also introduce two novel interaction techniques that engage users in anomaly detection:

1. Quantiles Brush - Specify the threshold that defines an anomalous event.
2. Samples Brush - Label anomalous events as normal events. Sometimes the system may trigger an event as anomalous because it has simply exceeded the specified threshold. However, users are allowed to label this event as normal (for whatever reason) so that similar events will not trigger false alarms in the future.

The authors also incorporated a Radial Graph to show real-time monitoring. This is basically an improved version proposed by the end users. This is illustrated in Figure 6. To enhance the visualization further, the authors design 3D model view of each workstation in the radial map to ease locating the faults. An example is shown in Figure 1 (G).

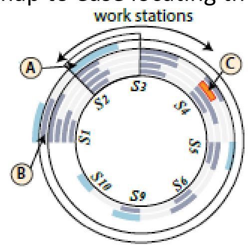

Figure 6: (A) Each concentric circle represents a product, the highlighted product is currently being processed on station s2. Light blue color represents ongoing processes on a station; (B) The length of a bar represents how long it takes for a part to finish its process on a station; (C) Fault occurs. From [Xu et al., Figure 3].

When selecting the design method for historical data analysis, the authors carefully justify many available options such as Gantt chart, Sankey diagrams, and MatrixWave, but finally deciding on using Marey's graph for historical data analysis after considering the pros and cons of each method.

The authors apply the method to identify the anomalous processes based on the samples provided by the end users. Both methods are able to capture and display the outliers clearly on the graph. This is an immediate approach to justify the design method.

The Radial graph to display real-time tracking data is essentially an improved version of Radial graph proposed by the end users. The redesign uses familiar visual metaphors and respects the mental model of the users about their assembly line. This will ensure better ease of understanding because the authors recognize that few of the end users (especially the operators) have advanced visual analytics applications.

5. Algorithm

The authors concisely cover the algorithmic details of Marey's graph for historical data analysis, Quantiles Brush and Samples Brush methods to engage user interactions in anomaly detection, as well as the Radial graph that tracks real-time data. One key design requirement is the ability to track assembly line performance in a real-time manner. The authors do not explicitly justify in their paper whether the algorithm is sufficiently fast to catch up with the in-coming stream of real-time data. It is only learned after the pilot run in the real assembly line that the design, and thus the algorithm, is able to meet the expectation of the end users via in-depth interviews and feedback gathering. No validation is performed at this level to measure time or memory performance.

6. Summary

Five validation methods were used in this paper as summarized in Figure 7.

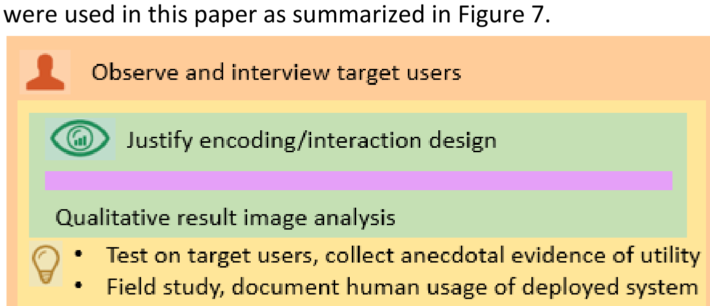

Figure 7: Validation approaches

7. Discussions and Conclusion

This paper explicitly covers all four design levels with validation at three of the levels. In conclusion, the visualization system is able to meet the requirements from the end users. The successful pilot run in the assembly line has provided the confidence to the users too.

However, there are few limitations and concerns for this visualization designs:

1. When display of longer time span is required in Marey's graph, all the polylines may appear as vertical lines which will affect the visibility of anomalies.
2. Currently the visualization system is designed to fit nicely on a screen with 1920 x 1080 resolution. More adaptive display layout should be considered. Being able to support larger screen may also help displaying Marey's graph with longer time span.
3. It is difficult to isolate sub-processes and parallel processes as they are currently overlaid in Marey's graph. More interactive design method can be considered in the future to enhance visibility.
4. The rationale behind using Radial graph is not clear and well technically justified. It is simply proposed by end users and enhanced by the authors. It is possible to explore other representation for better justifications.
5. In order to merge this visualization system into a big data environment, the infrastructure design (cloud, memory, CPU, storage, internet speed, etc.) is equally important. This paper does not explicitly address this concern.

8. References

[1] Panpan Xu, Honghui Mei, Liu Ren, Wei Chen, ViDX: Visual Diagnostics of Assembly Line Performance in Smart Factories. IEEE Transactions on Visualization and Computer Graphics, 23 (1): 291–300, Jan 2017.
[2] K. Matkovic, H. Hauser, R. Sainitzer, and M. E. Groller. Process visualization with levels of detail. In Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on, pp. 67–70, 2002.
[3] J. Jo, J. Huh, J. Park, B. Kim, and J. Seo. Livegantt: Interactively visualizing a large manufacturing schedule. IEEE Transactions on Visualization and Computer Graphics, 20(12):2329–2338, 2014.