VR-EDStream+EDA: Immersively Visualizing and Animating Event and Data Streams and Event-Driven Architectures in Virtual Reality

Roy Oberhauser[0000-0002-7606-8226]
Computer Science Dept.
Aalen University
Aalen, Germany
e-mail: roy.oberhauser@hs-aalen.de

***Abstract*—With increasing digitalization, the importance of data and events, which comprise its most fundamental level, cannot be overemphasized. All types of organizations, including enterprises, business, government, manufacturing, and the supporting IT, are dependent on these fundamental building blocks. Thus, evidence-based comprehension and analysis of the underlying data and events, their stream processing, and correlation with enterprise events and activities becomes vital for an increasing set of (grassroot or citizen) stakeholders. Thus, further investigation of accessible alternatives to visually support analysis of data and events is needed. This paper contributes VR-EDStream+EDA, a solution for immersively visualizing and interacting with data and event streams or pipelines and generically visualizing Event-Driven Architecture (EDA) in Virtual Reality (VR). Our realization shows its feasibility, and a case-based evaluation provides insights into its capabilities.**

Keywords - virtual reality; event-driven architecture; data pipelines; event stream processing; data stream processing; software architecture; visualization.

I. INTRODUCTION

It is said that "data is the new oil", with data playing a fundamental role in the digitalization and automation in various organizations, including enterprises, business, government, manufacturing, and IT. Events (a.k.a. record or message) are specific data consisting of a record of an occurrence. Modern software architectures, in the form of microservices or other decoupled reactive apps, are often event-driven, and microservice adoption in enterprises is growing, with IDC reporting 77% and GitLab reporting 71% of organizations (partially) using microservices [1][2].

The size of software applications has grown in size and complexity over the years and decades, as has the number of different apps or services and their interdependence or coupling in enterprises. Enterprise Service Buses (ESBs) are one example of how different apps and services can be coupled with each other without them even being aware of it. For example, it is said 57% of enterprises use between 1000-5000 business applications [3]. Consequently, more coupling and software reuse of (micro)services results in additional coupling and additional data and event traffic. At the business or enterprise level, each execution of an activity within a business process leaves a digital footprint of process-related events and the timepoint of execution, typically contained in various log files across the various IT systems or services. Analogous to the increase in and monitoring of network traffic, where network analysis supports analysis down to the packet level, an application- and tool-independent capability for equivalent data or event stream analysis is thus requisite.

Furthermore, as digitalization expands, various stakeholders (besides IT operators) may desire insight into the interactions between software and any related data and event processing. For example, developer responsibilities are expanding to include deployment, automation, performance management, user experience, and security, and increasingly responsible for the entire lifecycle of application development and operations [4].

Visualization and analysis of large dynamic data and event sets and their relations remains a challenge. While various specialized data tooling is available, alternative accessible generic (tool-independent) approaches for immersively visualizing and analyzing such (data or event) streams has not been sufficiently investigated. Furthermore, as data and processes become more relevant to the digital enterprise and stakeholders become more digitally savvy (grassroots or enterprise citizens), it is all the more relevant and challenging to include non-expert stakeholders in such data and event analysis. By leveraging Virtual Reality (VR), data and event analysis can be made more accessible to a wider set of stakeholders beyond the more specialized data scientists or software developers.

In prior VR-related work, in the area of processes we developed VR-BPMN [5] to visualize Business Process Modeling Notation (BPMN) models, while VR-ProcessMine [6] addressed process mining. In the area of Enterprise Architecture (EA), VR-EA [7] contributed a VR solution for ArchiMate EA models, VR-EAT [8] presented a VR-based solution for integrating dynamically-generated EA tool diagrams in VR, while VR-EA+TCK [9] integrated enterprise content and knowledge management systems in VR. In the software architecture and software engineering area, VR-UML [10] supports the Unified Modeling Language (UML) and VR-SysML [11] supports the Systems Modeling Language (SysML), while VR-Git [12] supports Git repositories. This paper contributes VR-EDStream+EDA, a solution for immersively visualizing and interacting with data, events, and generically visualize EDA in VR. Our prototype realization shows its feasibility, and a case-based evaluation provides insights into its capabilities for addressing the aforementioned challenges.

The remainder of this paper is structured as follows: Section 2 discusses related work. In Section 3, the solution is described. Section 4 provides details about the realization. The evaluation is described in Section 5 followed by a conclusion.

II. RELATED WORK

Our search for related work includes the data visualization survey by Qin et al. [13] only mention events streams with regard to SQL-like query support. A survey on immersive analytics by Fonnet and Prié [14] includes no citations related to streams, and only two related to events: IDEA [15], which depicts user activity logs in a 3D cylindrical scatterplot while tracking a mobile chair, and DebugAR [16], which uses Augmented Reality (AR) for debugging.

As to immersive toolkits, the DXR toolkit [17] offers support for building immersive visualizations, and does not mention events nor streams. IATK [18] is another immersive analytics toolkit, whereby events, messages, and streams are not mentioned nor addressed. Stream [19] uses head-mounted AR devices to support visual data analysis. Spatially-aware tablets are used for interaction and input. In contrast, our solution does not necessitate additional AR hardware or a real tablet, since a virtual VR tablet is provided. Furthermore, our solution does not require or utilize individual linked 2D scatter plots. This would potentially impede scalability depending on the connectedness and grouping of the nodes involved.

Reactive Vega [20] is a streaming dataflow architecture that supports declarative interactive visualization. Its architecture and parser are implemented in JavaScript, and intended to run in a web browser or with Node.js. Popular tools for visualizing event systems, such as Kafka and RabbitMQ, include the web applications Grafana and Kibana, or some tool implementation in combination with D3.js.

In contrast to the above, VR-EDStream+EDA provides a generic (application and service independent, event platform independent, and programming language independent) approach for immersive event and data stream visualization and animation in VR.

III. SOLUTION

VR is a mediated simulated visual environment in which the perceiver experiences telepresence. VR provides an unlimited space for visualizing a growing and complex set of enterprise models and processes and their interrelationships simultaneously in a spatial structure. As the importance, scale, inter-dependence, and coupling of data and events for IT infrastructure grows, an immersive environment can provide an additional visualization capability to comprehend and analyze both the structurally complex and interconnected static relations and the dynamic interactions between digital elements.

In support of the possible benefits of an immersive VR experience vs. 2D for performing an analysis task, Müller et al. [21] investigated a software analysis task that used a Famix metamodel of Apache Tomcat source code dependencies in a force-directed graph. They found that VR does not significantly decrease comprehension and analysis time nor significantly improve correctness (although fewer errors were made). While interaction time was less efficient, VR improved the UX (user experience), being more motivating, less demanding, more inventive/innovative, and more clearly structured.

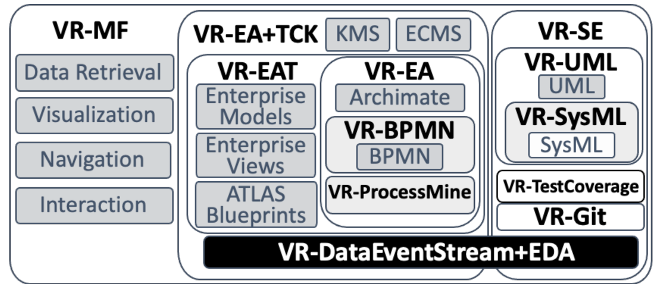

Figure 1. The VR-EDStream+EDA solution concept (black) in relation to our prior VR solution concepts.

To provide a context and background for our generalized solution concept for VR-EDStream+EDA we refer to Figure 1. VR-EDStream+EDA utilizes our generalized VR Modeling Framework (VR-MF) [6], which provides a VR-based domain-independent hypermodeling framework, which addresses four primary aspects that require special attention when modeling in VR: visualization, navigation, interaction, and data retrieval. VR-EA [6] provides specialized direct support and mapping for EA models in VR, including both ArchiMate as well as BPMN via VR-BPMN [5]. VR-ProcessMine [6] provides support for (business or software) process mining in VR. VR-EAT [8] extends this further with integration of EA tools for accessing dynamically generated diagrams and models from an EA tool in VR. VR-EA+TCK [9] extends these capabilities by integrating further enterprise knowledge, information, and content repositories such as a Knowledge Management Systems (KMS) and Enterprise Content Management Systems (ECMS). Since data streams, event streams, and EDA involve data and/or software and inter-software communication, VR-EDStream+EDA (shown in black) spans both the enterprise and software engineering areas, applicable to relevant stakeholders depending on their focus and intention.

A. Visualization in VR

A generic way of portraying an EDA or dynamic stream of data (records or packets) or events is as a set of nodes and a Directed Acyclic Graph (DAG) to indicate the producer (source) and consumer (sink), as exemplified in Figure 2. Events (messages) can be grouped and stored in topics, accessible to multiple producers or consumers. A 3D sphere is used to depict nodes, a 3D empty pipe (straw) is used for the graph, and the event is depicted as a 3D capsule that is dynamically animated within the pipe.

In the immersive space of VR, analysis is affected by the distance of objects, thus ideally an initial automatic placement should place them in relative proximity to avoid delays due to traveling to objects to interact with them. While a force-directed graph rebalances the distance of object automatically, it takes time to reach a steady state, while any manual element replacements by a user can cause side-effects. Inspired by 2D chord diagrams used in visual data analytics, we considered how to use the third dimension to reduce clutter, reduce connector collisions, and retain order and legibility. Thus, to support scalability while minimizing the collision of connectors, nodes are initially placed on the outer edge of an imaginary sphere, while node groups follow along a planar circle on the sphere's edge as shown in Figure 3. The largest sized group (based on number of nodes) is placed near the equator and serves as the basis for the sphere circumference, while smaller groups are placed accordingly closer to the poles. This grouping creates an implicit layering effect. Nodes in the same group have the same color, and the size of a node (sphere) is dependent on the number of connectors (streams), with the smallest having none.

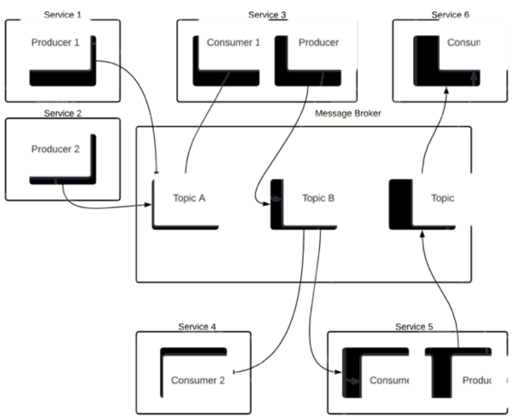

Figure 2. Example EDA couplings between services.

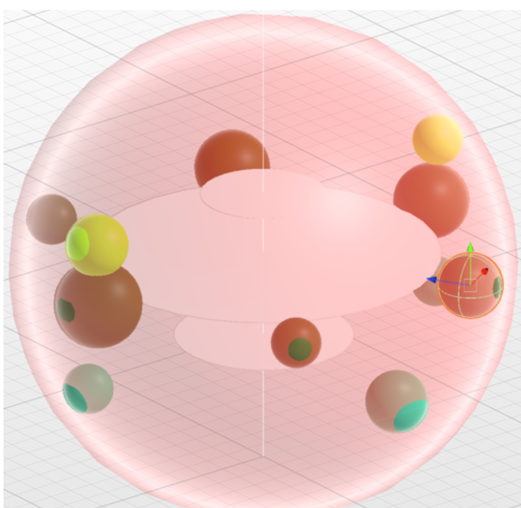

Figure 3. Node placement on spherical edge with groups on planar circles.

Figure 4. Event stream portrayal in VR: nodes as spheres (left arrow), semitransparent tube as stream (right arrow), and animated capsule as event (middle arrow).

To depict a stream, transmission, or processing of events or data in VR, a semi-transparent tube is used with nodes portrayed as spheres on both ends, and an animated capsule indicating the direction of source and sink, shown in Figure 4.

B. Navigation in VR

The immersion afforded by VR requires addressing how to navigate the space while reducing the likelihood of potential VR sickness symptoms. Thus, two navigation modes are included in the solution: the default uses gliding controls, enabling users to fly through the VR space and view objects from any angle they wish. Alternatively, teleporting permits a user to select a node or event and be instantly placed there (i.e., by instantly moving the camera to that position); while this can be disconcerting, it may reduce the susceptibility to VR sickness for those prone to it that can occur when moving through a virtual space.

C. Interaction in VR

Elements in the model can be freely moved. Since interaction with VR elements has not yet become standardized, in our VR concept, user-element interaction is handled primarily via the VR controllers and a virtual tablet. Our VR-Tablet provides detailed context-specific element information, and can provide a virtual keyboard for text entry fields (via laser pointer key selection) when needed.

D. Capabilities

Solution capabilities include:

- A network-based mechanism for monitoring and collecting data or events
- A common data and event storage mechanism
- EDA definition and configuration
- Node grouping, placement, naming, coloring
- Store and load VR model changes
- Define event flow time period
- Dynamic event flow step and speed control

IV. REALIZATION

As a realization of our solution concept, our prototype is partitioned into a data hub, a backend for data processing, and a front end responsible for VR visualization.

The data hub was implemented as a microservice. For storage, the InfluxDB was used as a database due to: 1) its time series support and 2) since its storage requirements were deemed significantly smaller for large time series datasets than any alternatives, a benefit when scaling the solution. For receiving events generically, a microservice RESTful interface for receiving JSON event or data records was realized in Python using the FastAPI web framework. In addition to the REST interface, Telegraf - part of InfluxData platform, offers a open source server-based agent written in Go for collecting and sending metrics and events from databases, systems, and sensors to the InfluxDB. Either interface can be flexibly used to extract or collect events, applying an interceptor, proxy, or decorator pattern as appropriate.

VR was implemented with Unity and SteamVR. The main classes involved are shown in the class diagram in Figure 5. The NodeManager manages the overall configuration and

 ISBN: 978-1-68558-082-7

portrayal of nodes and edges and passes events to the nodes. EventManager computes events based on the data records, and the NodeManager passes these to the appropriate Node, which generates and fires a visual capsule. Timeline manages the animation including timepoint and speed of events. DataHubManager is used for accessing the datasets.

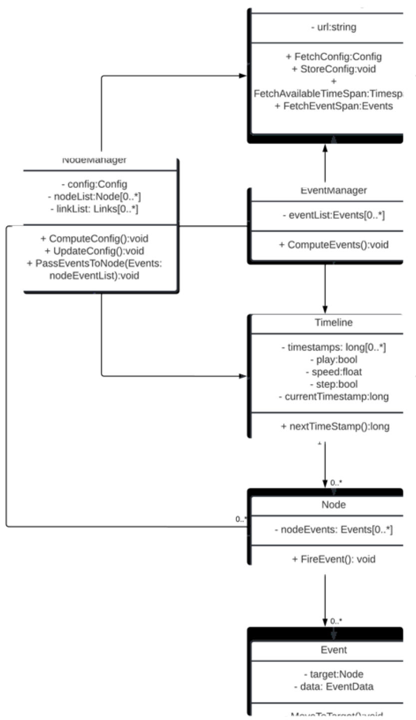

Figure 5. Class diagram for VR visualization support.

Integration with two different event systems was performed. Apache Kafka is an open-source distributed event streaming platform implementing the message broker pattern. Kafka Connect supports data integration between databases, key-value stores, search indexes, and file systems. The connectors receive and transmit data to and from topics as a source or sink, and various extensible implementations are available. Examples include a Source Connector that streams database updates to a topic, collect server metrics to a topic, forward topic records to Elasticsearch, etc. Confluent ksqlDB was used in the test applications as a database supporting queries in SQL syntax for stream processing applications based on Kafka Streams. To ensure a generic solution, a second popular publish/subscribe message broker event system, RabbitMQ, was also utilized in the evaluation. For more details and a comparison of these distributed event systems, we refer to Dobbelaere & Esmaili [22].

Metainformation collected via REST or Telegraf and retained in the database with each record are as follows: source, target, timestamp, payload. Thus, the payload can be data, an event, a message, etc. If no target exists, then any null or fake named node can be used (equivalent to a null device in Unix).

For interaction support with the data, a VR-Tablet menu offers these display modes:

- Animated Timeline for controlling dynamic stored or real-time playback (Figure 6 left),
- Querying the event or data store (Figure 6 right),
- Color customization (Figure 7),
- Object Details for a selected node (Figure 8) or capsule (i.e., event or data record, Figure 9), and
- Settings for storing and fetching configurations (not shown).

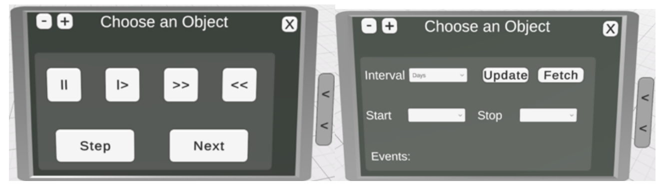

Figure 6. Dynamic animation interface (left) and Query interface (right).

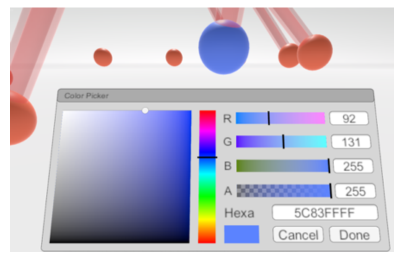

Figure 7. Object color customization.

Figure 8. Node detail interface after node selection.

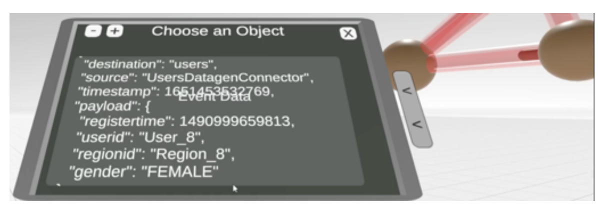

Figure 9. Example event details after selecting red capsule.

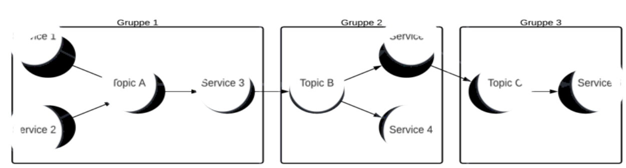

Figure 10. Abstracted node grouping EDA example.

Configurations stored in JSON format permit stakeholders to flexibly group nodes and streams, in essence defining the EDA (e.g., based on microservices) the way they wish based on their interst. An example cross-service EDA is shown in Figure 10. Nodes in a group are assigned the same color.

V. EVALUATION

The evaluation of the solution concept is based on design science method and principles [23], in particular, a viable artifact, problem relevance, and design evaluation (utility, quality, efficacy). For this, a case study based on scenarios is used, as our evaluation focuses on visual support for a variety of generic EDA configurations via node groupings and streams. These visual scenarios consisted of various node grouping and coupling configuration.

A. Event System Integration Tests

For generating event data for the evaluation, the Confluent Quickstart Demo using ksqlDB in combination with Kafka Connect was used with two connectors to the topics pageviews und users. A second configuration based on Confluent Kafka consisted of one producer and three consumers in Python. To ensure the solution was not Kafka dependent, a third configuration using only RabbitMQ with our microservice was also tested.

B. Single Large Group Connected to One Node

As a scalability scenario, a single group of 100 nodes all connected is shown in Figure 11. Note that in VR, due to its unlimited space, there are no actual limitations in navigating to nodes and comprehending large models.

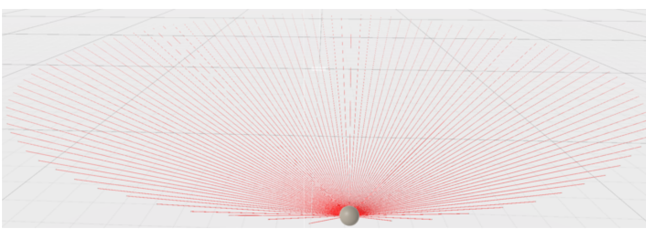

Figure 11. Scalability test: a group of 100 nodes connected to one node.

C. Unbalanced Groups Randomly Interconnected

This scenario consisted of three unbalanced groups: one group with 20 randomly intra-connected nodes, and two inter-connected groups consisting of a single node each, as portrayed in Figure 12. Note each group has a different node color, and more connected nodes are larger, and smaller groups are near the poles of the sphere, with the largest group at the equator.

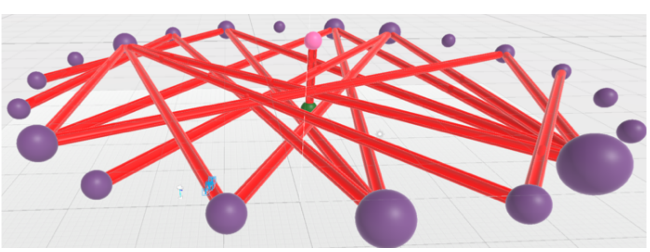

Figure 12. Three groups: one with 20 randomly intra-connected nodes and two inter-connected groups consisting of a single node each.

D. Multiple Balanced Highly Interconnected Groups

In this scenario, three balanced groups of 20 nodes each are randomly inter- and intra- connected with other nodes, as shown in Figure 13.

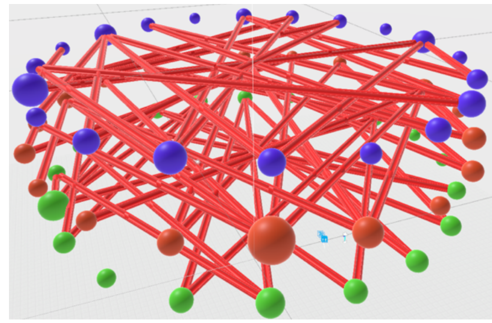

Figure 13. Three groups of 20 nodes each with random coupling.

E. Multiple Unbalanced Groups Irregularly Interconnected

To test many unbalanced groups with different degrees of connectedness, this scenario had five groups, one group with 20 nodes and the rest consisting of 5-10 nodes with random unbalanced coupling. The result is shown in Figure 14.

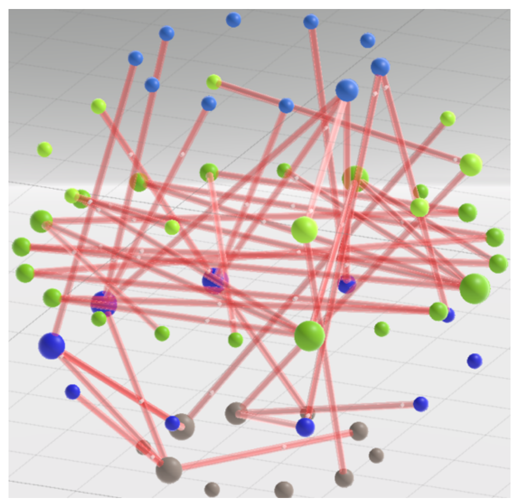

Figure 14. Five groups (with 20 and 5-10 nodes) and random coupling.

F. Discussion

The scenarios with their various node grouping configurations show various possibilities and capabilities of the generic solution concept. Our concept generically simplifies the understanding of complex software systems for those stakeholders only concerned with event and data flow. It does this by immersively depicting sources and sinks as nodes in a spatially compact (3D spherical) layout, and animating any interaction between them.

flows and communication streams for data and events by focusing only on the essential and removing all else, yet is scalable and immersive. By immersively visualizing these key aspects, various (grassroot) stakeholders can now access and comprehend the flow of event data in an animated fashion. The default placement of an EDA configuration provides a

 ISBN: 978-1-68558-082-7

starting point for analysis, and users can move and recolor nodes as desired, and query applicable datasets.

VI. CONCLUSION

Access to and comprehension and analysis of the underlying events and data, their stream processing, and correlation with enterprise events and activities will become increasingly vital for a larger set of stakeholders. This paper contributed VR-EDStream+EDA, a VR solution for immersively visualizing and interacting with event and data streams or pipelines and EDA. Our realization showed its feasibility. A case-based evaluation provided insights into its capabilities, showing its ability to deal with balanced and unbalanced node group configurations and various coupling scenarios. Its customizable node placement in addition to the event animation provides an immersive visualization alternative for various stakeholders to comprehend the dynamic event and data flow data that form the basis of the IT system interactions in enterprises.

Future work includes adding additional data analyses, integration with our other enterprise hypermodeling VR solutions, an interview study with experts, and a comprehensive industry empirical study.

ACKNOWLEDGMENT

The authors thank Markus Harder for his assistance with the design, implementation, evaluation, and diagrams.

REFERENCES

[1] M. Loukides and S. Swoyer, “Microservices Adoption in 2020,” O’Reilly Media, Inc., 2020. [Online]. Available from: https://www.oreilly.com/radar/microservices-adoption-in-2020/ 2023.03.25

[2] GitLab, “A Maturing DevSecOps Landscape,” 2021. [Online]. Available from: https://about.gitlab.com/images/developer-survey/gitlab-devsecops-2021-survey-results.pdf 2023.03.25

[3] “New Ponemon Study Reveals Application Security Risk At All Time High: 1 in 2 Enterprises Need Better Protection,” 2015. [Online]. Available from: https://www.businesswire.com/news/home/20151210006098/en/New-Ponemon-Study-Reveals-Application-Security-Risk-At-All-Time-High-1-in-2-Enterprises-Need-Better-Protection 2023.03.25

[4] M. Shirer, “IDC Survey Illustrates the Growing Importance of Developers to the Modern Enterprise,” IDC, 2021. [Online]. Available from: https://www.idc.com/getdoc.jsp?containerId=prUS48058021 2023.03.25

[5] R. Oberhauser, C. Pogolski, and A. Matic, "VR-BPMN: Visualizing BPMN models in Virtual Reality," In: Shishkov, B. (ed.) Business Modeling and Software Design (BMSD 2018), LNBIP, vol. 319. Springer, Cham, 2018, pp. 83–97, doi.org/10.1007/978-3-319-94214-8_6.

[6] R. Oberhauser, "VR-ProcessMine: Immersive Process Mining Visualization and Analysis in Virtual Reality," the Fourteenth International Conf. on Information, Process, and Knowledge Management (eKNOW 2022), IARIA, 2022, pp. 29-36.

[7] R. Oberhauser and C. Pogolski, "VR-EA: Virtual Reality Visualization of Enterprise Architecture Models with ArchiMate and BPMN," In: Shishkov, B. (ed.) Business Modeling and Software Design (BMSD 2019), LNBIP, vol. 356, Springer, Cham, 2019, pp. 170–187, doi.org/10.1007/978-3-030-24854-3_11.

[8] R. Oberhauser, P. Sousa, and F. Michel, "VR-EAT: Visualization of Enterprise Architecture Tool Diagrams in Virtual Reality," In: Shishkov B. (eds) Business Modeling and Software Design (BMSD 2020), LNBIP, vol 391, Springer, Cham, 2020, pp. 221-239, doi.org/10.1007/978-3-030-52306-0_14.

[9] R. Oberhauser, M. Baehre, and P. Sousa: VR-EA+TCK: Visualizing Enterprise Architecture, Content, and Knowledge in Virtual Reality. In: Shishkov, B. (eds) Business Modeling and Software Design (BMSD 2022), LNBIP, vol 453, Springer, Cham, 2022, pp. 122-140, doi.org/10.1007/978-3-031-11510-3_8.

[10] R. Oberhauser, “VR-UML: The unified modeling language in virtual reality – an immersive modeling experience,” International Symposium on Business Modeling and Software Design (BMSD 2021), Springer, Cham, 2021, pp. 40-58, doi.org/10.1007/978-3-030-79976-2_3

[11] R. Oberhauser, “VR-SysML: SysML Model Visualization and Immersion in Virtual Reality,” International Conference of Modern Systems Engineering Solutions (MODERN SYSTEMS 2022), IARIA, 2022, pp. 59-64.

[12] R. Oberhauser, “VR-Git: Git Repository Visualization and Immersion in Virtual Reality,” The Seventeenth International Conference on Software Engineering Advances (ICSEA 2022), IARIA, 2022, pp. 9-14.

[13] X. Qin, Y. Luo, N. Tang, and G. Li, “Making data visualization more efficient and effective: a survey,” The VLDB Journal, 29, pp.93-117, 2020.

[14] A. Fonnet and Y. Prié, "Survey of Immersive Analytics," in IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 3, pp. 2101-2122, 2021.

[15] A. Fonnet, F. Melki, Y. Prié, F. Picarougne, and G. Cliquet, “Immersive Data Exploration and Analysis,” Student Interaction Design Research Conference, Helsinki, Finland, hal-01798681, 2018, https://hal.science/hal-01798681.

[16] P. Reipschläger et al., “DebugAR: Mixed dimensional displays for immersive debugging of distributed systems,” In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1-6.

[17] R. Sicat et al., “DXR: A toolkit for building immersive data visualizations,” IEEE transactions on visualization and computer graphics, 25(1), 2018, pp.715-725.

[18] M. Cordeil et al., "IATK: An Immersive Analytics Toolkit," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 200-209, doi: 10.1109/VR.2019.8797978.

[19] S. Hubenschmid, J.Zagermann, S. Butscher, and H. Reiterer, “Stream: Exploring the combination of spatially-aware tablets with augmented reality head-mounted displays for immersive analytics,” Proc. 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1-14.

[20] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, "Reactive Vega: A Streaming Dataflow Architecture for Declarative Interactive Visualization," In: IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 1, pp. 659-668, 2016, doi: 10.1109/TVCG.2015.2467091.

[21] R. Müller, P. Kovacs, J. Schilbach, and D. Zeckzer, "How to master challenges in experimental evaluation of 2D versus 3D software visualizations," In: 2014 IEEE VIS International Workshop on 3Dvis (3Dvis), IEEE, 2014, pp. 33-36.

[22] P. Dobbelaere and K.S. Esmaili, “Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations,” In: Proc. 11th ACM int’l conference on distributed and event-based systems, 2017, pp. 227-238.

[23] A.R. Hevner, S.T. March, J. Park, and S. Ram, “Design science in information systems research,” MIS Quarterly, 28(1), 2004, pp. 75-105.