Database Model Visualization in Virtual Reality: A WebVR and Benediktine Space Approach

Roy Oberhauser
Computer Science Dept.
Aalen University
Aalen, Germany
e-mail: roy.oberhauser@hs-aalen.de

Abstract - **As the data era and the Internet of Everything unfolds, databases are becoming ubiquitous and an integral part of software while alternative database types such as NoSQL grow in popularity. Thus, software engineers, not just database specialists, are more likely to encounter and need to deal with these databases. While Virtual Reality (VR) technology has increased in popularity, its integration in software development tooling has been limited. One application area for WebVR technology includes database-model visualization, permitting web-based cross-platform and remote VR access. This paper describes Virtual Reality Immersion in Data Models (VRiDaM), a generic database-model approach for visualizing, navigating, and conveying database-model information interactively in a web browser using WebVR technology utilizing Benediktine space visualization for heterogeneous (relational and NoSQL) database models. A prototype shows its viability and an empirical study looked at usability, effectiveness, and efficiency.**

Keywords - virtual reality; database visualization; database tools; visual development environments; database modeling; software engineering; WebVR; Benediktine space.

I. Introduction

Data has become the most coveted "raw material" of both our time and the foreseeable future, in some respects analogous to a gold rush. Cisco estimates there will be 27bn networked devices by 2021 [1]. The ongoing digitalization involving Industry 4.0 and the Internet of Everything will imply a large increase in databases to be able to store and retrieve this data, in particular embedded databases. IDC estimates the current annual data creation rate at 16.1ZB (Zettabytes), and by 2025 163ZB, with embedded data by then constituting nearly 20% of all data created [2]. As data explodes, software engineers are increasingly faced with the daunting task of structuring and analyzing such databases across various DataBase Management System (DBMS) types, including relational and NoSQL types such as document (semi-structured), key-value, wide column (extensible record), in memory, and graph [3].

Thus, software engineers are increasingly faced with developing and maintaining software that integrates some form of data repository, data store, or database. While the original developers may have a clear (and correct to a certain degree) mental model of their actual data model, the maintenance situation is exacerbated by proliferation of relational (mostly SQL) and NoSQL database types and the relatively high turnover rates common in the software industry, resulting in developers unfamiliar with the data models attempting to quickly comprehend the database structures involved with these software applications.

Humans tend to be visually-oriented and can detect and remember visual patterns well. Information visualization has the potential to support human understanding and insight while dealing with resource constraints on the human as well as computer side. Common ways for visually conveying database structures include 2D entity-relationship (E-R) modeling and diagrams [4], but these are typically applied to relational databases (RDB) and NoSQL databases may or may not have a tool that includes visual support. As to DBMS tools, often a database product has a preferred product-specific tool offering web-based or standard 2D graphical user interfaces (GUIs), while certain tools support multiple database products of one specific type (e.g., MySQL workbench for SQL databases).

Contemporaneously, VR has made inroads in its accessibility as hardware prices have dropped and capabilities have improved. The VR market is rapidly expanding, with VR revenue reaching $2.7bn in 2016, a 10-fold increase to $25bn by 2021 [5]. However, software engineers mostly do not have access to integrated VR capabilities in their development tools. Broad VR usage is relatively new and this market segment small compared to the general VR market.

This application paper contributes Virtual Reality Immersion in Data Models (VRiDaM, pronounced like freedom), a generalized approach to heterogeneous (relational and non-relational) database-model visualization in VR , using WebVR in a web browser and a Benediktine-space [6]-[9] visualization paradigm. Thus, database models from different data store types can be visualized and navigated (locally or remotely) in VR via a cross-platform web browser and a VR headset and controller. We describe its principles and features for visualizing, navigating, and conveying database information interactively to support exploratory, analytical, and descriptive cognitive processes [10]. A prototype implementation demonstrates its viability and its usability is evaluated in an empirical study.

The paper is organized as follows: the following section discusses related work; Section 3 presents our solution approach. In Section 4, our implementation is described. An evaluation is described in Section 5, followed by a conclusion.

 ISBN: 978-1-61208-668-2

II. Related Work

We are unaware of directly related work regarding database visualization using VR. VR Juggler [11] provides VR support for developing VR applications, but not for database modeling and visualization. As to VR approaches for software structure visualization, ExplorViz [12] is a WebVR application that supports VR exploration of 3D software cities using Oculus Rift together with Microsoft Kinect for gesture recognition. As to non-VR visualization, [13] provides an overview and survey of 3D software visualization tools across various software engineering areas. Software Galaxies [14] gives a web-based visualization of dependencies among popular package managers and supports flying, with each star representing a package clustered by dependencies. CodeCity [15] is a 3D software visualization approach based on a city metaphor and implemented in SmallTalk on the Moose reengineering framework. X3D-UML [16] provides 3D support with UML grouping classes in planes. In contrast, VRiDaM focuses on visualizing database structures and leverages WebVR capabilities without requiring gestures.

Database management (DBM) tools are typically DB type-specific and require some installation. Each professional RDB product usually offers its own tool, but since most RDBs support the Structured Query Language (SQL), certain RDB tools can access other RDBs using RDB-specific drivers. For example, MySQL Workbench works across multiple databases and supports reverse-engineering of 2D E-R diagrams. Schemaball [17] provides a 2D circular composite schema diagram for SQL databases. As to 3D RDB tools, DIVA (Database Immersive Visual Analysis) uses stacked rings with rectangular tables attached to them (forming a cylinder), with the tables with the most foreign keys sorted to the top of the stack. Alternatively, stacked square layers of tables can be displayed and 2D E-R diagrams shown. Actual data values are not visualized. For NoSQL databases, each database type differs and the associated DBM tools. One example of a popular wide-column database (WDB) is Apache Cassandra, for which DataStax Studio is a Java-based DBM tool with a web GUI (Graphical User Interface). As to document-oriented databases (DDBs), MongoDB is a popular example and MongoDB Compass, Robomongo, and Studio 3T being example DBM tools. For graph databases (GDB), Neo4j is popular and graph DBM tools include Neo4j admin, Structr, Gephi, Graffeine, etc. In contrast, the VRiDaM approach is more generalized to work across heterogeneous DB types, thus permitting users to only ramp up on one tool, it is cross-platform and provides an immersive web-based VR experience. Furthermore, in contrast to the 3D DIVA or 2D Schemaball, our approach avoids the visual connection 'yarnballs' as the number of connections and tables scale.

Work on big data visualization techniques in conjunction with VR is discussed by Olshannikova et al. in [18]. Herman, Melançon, and Marshall [19] survey work on graph visualization and navigation for information visualization. In contrast, our focus is displaying the database-model structure, not on displaying and analyzing large amounts of data per se, and we apply Benediktine spatial placement in conjunction with a dynamic self-organizing force-directed graph [20].

III. Solution

Visualization, especially VR with its wide viewing angles, can leverage peripheral vision for information, whereby visual data is both consciously and unconsciously seen and processed. If leveraged well, visualization can be cognitively easier in providing insights than an equivalent textual analysis would require. Traditional text-centric tabular formats or boxes in E-R diagrams do not explicitly take advantage of such visual capabilities. Also, if the contents of each database table were visualized as a rectangular 2D object, as it scales both in number of tables and the size of various tables, lucidity issues occur that nullify the advantage of VR visualization.

To provide some background context for our solution, we describe several perspectives that were considered. Information can be grouped and large amounts of information provided in a relatively small amount of graphical space. Yet humans are limited in their sensory perception and focus, and thus visualization considerations include: determining the proper balance for what to place into visual focus in which context, what to place into the periphery, what to hide or show, and to what extent and at what point should what be visualized. To develop actionable insights from visualization, the knowledge crystallization cognitive process involves acquiring information, making sense of it, creating something new, and acting on it [21]. Regarding visual perception, gestalt psychology [22] is based on the four principles of emergence, reification, multistable perception, and invariance. Formulated gestalt grouping laws regarding visual perception include proximity, similarity, closure, symmetry, common fate, continuity, good gestalt, past experience, common region, and element connectedness. For visual representation, we considered Don Norman's design principles, in particular affordance, consistency, and mapping [23]. Other concepts considered were [9] information space, cognitive space, spatialization, ordination, and pre-attentive processing, which refers to the accumulation of information from the environment subconsciously [24]. Visualization techniques explicitly analyzed with regard to their appropriateness for displaying data models in VR included books, cone trees, fisheye views, information cubes, perspective walls, spheres, surface plots/cityscapes/3D bar graphs, viewpoints, workspace/information space/3D rooms, worlds in worlds, and Benediktine space [19][21].

A. Benediktine Space

Benediktine space transforms or maps an information object's attributes to extrinsic (e.g., Cartesian coordinates, time) and intrinsic (e.g., size, shape, color) information spatial dimensions. To keep objects from overlapping, mapping principles include exclusion, maximal exclusion, scale, and transit [6][7][8][9].

B. WebVR

WebVR is a Mozilla JavaScript API that enables web browsers to access VR hardware. A-Frame is an open source framework for displaying VR content within HTML. It is based on an entity component system architecture in which each object in a scene is an entity (a container) consisting of components that provide desired functionality or behavior for that entity. A-Frame uses the three.js library to provide 3D graphics display in the browser and simplify WebGL programming. Systems are data containers. In contrast to game or PC station VR solutions, the use of VR within web browsers is relatively new, thus in deciding on the visualization techniques to use we considered the limitations and available capabilities and performance offered with WebVR for standard hardware (such as notebooks) that developers might use. Visualization considerations included selecting what and how many objects are displayed at any given time. Furthermore, in contrast to games, there is no real upper limit on the number of simultaneous entities a database or database model may have, which may overtax the computing and rendering capabilities and have negative impacts on the frame rates in VR, making the experience unsatisfactory and possibly resulting in VR sickness.

To characterize WebVR performance, we experimented with the implementation, some measurements of which are shown in Table 1. They are averaged across 10 measurements for 10 tables with 50 columns each on a notebook with Intel Core i5-3320M 2.6Ghz, 8GB RAM, Win7 x64, Intel HD Graphics 4000, Chrome 60.0.3112.113 and A-Frame 0.6.1.

TABLE I. AVERAGE A-FRAME PERFORMANCE (FRAMES PER SECOND)

Variants	Loading (fps)	Running (fps)
spheres, no edges	25	61
spheres, with edges	21	53
labeled spheres, no edges	11	19
circles, no edges	25	57
spheres, no edges (10x nodes)	3	12

Based on our experience and measurements with the A-Frame implementation, the following performance findings were made and affected our solution: 1) the number of rendered objects has a major impact on performance, 2) edges have a negative effect on performance, 3) text labels have a severe impact, 4) circles and spheres are equivalent.

For Finding 1, that implies that only the minimum number of objects should be displayed. Thus, rows (data values) will only be shown for selected column, not for all columns. Due to Finding 2, objects will be displayed without edges and connectors between objects will be avoided (force-graph). Due to Finding 3, text will be limited and the data value only shown when a circle (tuple) is selected.

C. VRiDaM Solution

Our VRiDaM solution architecture is based on the information visualization reference model by Card et al. [25] (see Figure 1) and involves transforming the raw data and its metadata to internal structures (the first two being purely data forms), and then mapping these to visual element structures, and transforming these to the views seen be the user (the last two being visual forms). To adjust the views, the user provides interaction input affecting one of the aforementioned transformation steps.

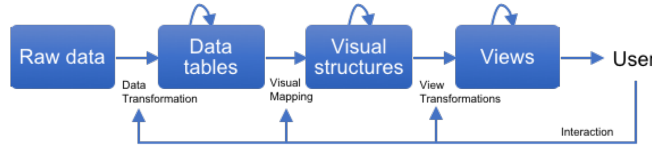

Figure 1. VRiDaM architecture.

The VRiDaM solution principles (P:) are as follows:

Support heterogeneous database-model types. Our VR approach works across different databases products and database types (SQL and NoSQL), thus familiarity with a single VR app could be leveraged across the various database types. Alternatively, currently developing unique VR app tools for each database and/or database type would be exorbitant relative to the number of software engineers that have VR capabilities and have database-model interests, and inefficient from a learning/training perspective.

Leverage spatial visualization in VR using a Benediktine spatial object placement approach. Our approach leverages the additional dimensional visualization and navigational capabilities VR provides (within current limitations of WebVR). Specifically, we utilize a Benediktine space visualization technique [6] with visual object spatial placement based on extrinsic spatial dimensions, whereas other entity properties are represented by intrinsic dimensions (form, size, color, etc.). The principle of exclusion ensures no two objects occupy the same spatial location, and the principle of maximal exclusion ensures that different data items are separated as much as possible [7].

Leverage dynamic self-organizing force-directed graph visualization to indicate the strength of relationship between objects via proximity. For visualizing relations, dynamic self-organizing force-directed graph placement [20] can be used in place of connectors to indicate via proximity more strongly related entities from those that are less- or unrelated. This is combined with visual highlighting of related objects when selecting an object. In this way we intend to avoid the "ball of yarn" issue with visual connectors as database models scale.

Cross-platform web-centric VR access. Our approach utilizes a browser-based implementation based on WebVR to enable cross-platform access to VR content assuming the user has a VR headset. Software engineers often work across different operating systems (Windows, Linux, etc.), and this permits them to utilize the app from any platform with appropriate WebVR browser support.

IV. IMPLEMENTATION

The WebVR-based prototype uses A-Frame and D3.js, which produces dynamic, interactive data visualizations in web browsers. For a self-organizing force-directed graph, our implementation uses the d3-force-3d physics engine from D3. Firefox and Chrome were used as web browsers. For database connectivity, the Spring Framework 4.3.1 was used and tested with PostgreSQL, MSSQL, MongoDB, Cassandra, and Neo4j. Content for the force-directed graph component was transformed to JSON format. The Northwind Trading sample database consisting of 13 tables and 6635 records was used, primarily, . Figure 2 shows the class diagram regarding database integration.

 ISBN: 978-1-61208-668-2

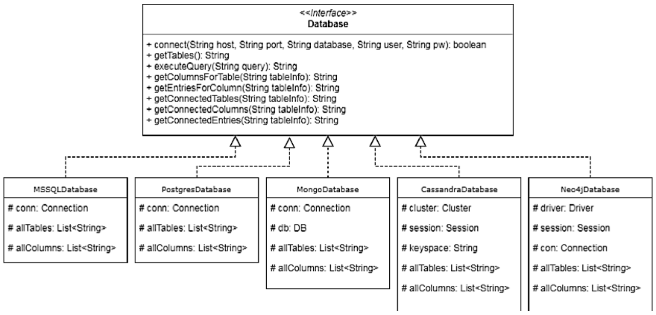

Figure 2. Class diagram of database integration.

The following visual object forms were selected:

Cubes are used to represent database tables (collections for document stores, or labels for graphs), analogous to cube furniture that can be used as a table (Figure 3).

Cylinders are used to represent database columns (set of similarly typed data, known as keys for document stores or graphs), analogous to columns in a building (Figure 4).

Planes are used for projecting the database data records (rows, tuples, or entries - the actual data values) since these can be very large in both columns and records and would thus occupy a large amount of VR space Figure 5). A plane supports maximum readability and permits VR navigation around it.

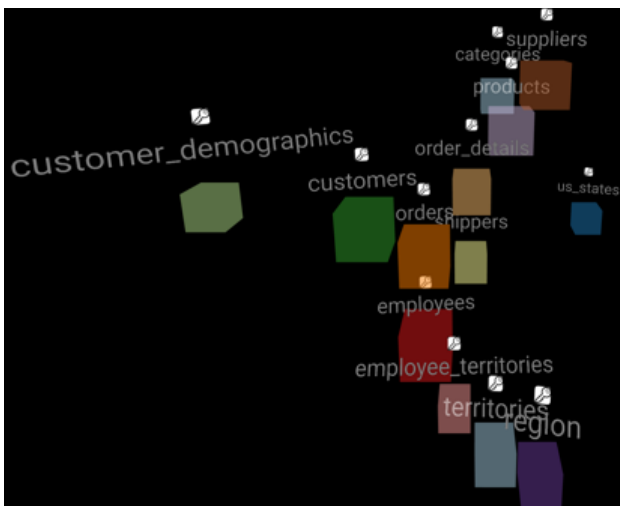

Figure 3. VRiDaM showing Northwind tables in Benediktine space.

Figure 4. Columns visible orbiting selected table (identical color).

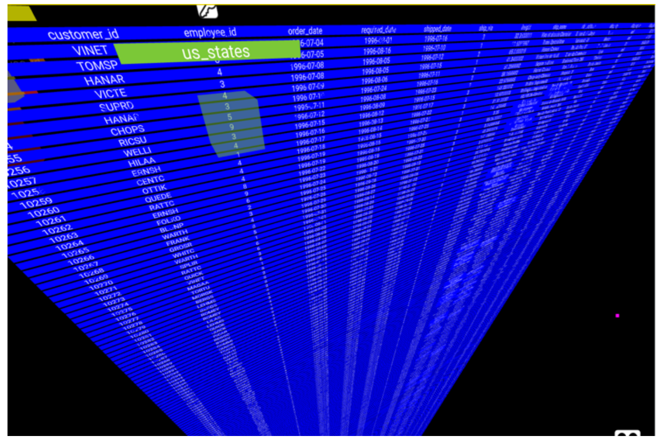

Figure 5. Table records projected onto a plane.

As to navigation and interaction, users can move objects as desired using standard VR controllers (we used an HTC Vive) or can use a mouse and keyboard. As seen in Figure 6, besides using spatial proximity to indicate closer associations, if a user selects an object, that object and all its directly referenced objects are highlighted. A key image is provided as an affordance and, if selected, a popup shows the primary and foreign keys names. If desired, lines can optionally be used to emphasize relations as shown in Figure 7.

Figure 6. Primary and foreign keys for table shown as popups.

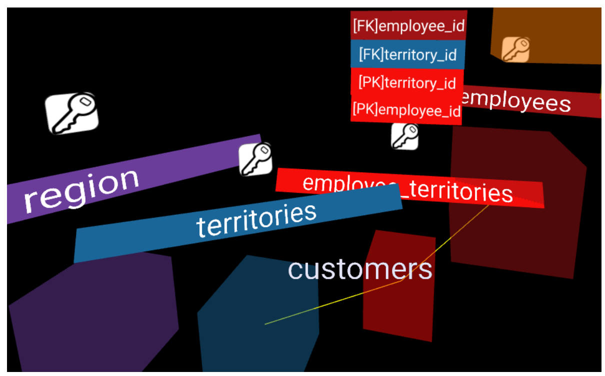

Figure 7. Example optional relation visualization using lines.

The configuration menu is overlaid and can be used to connect to a database and query (e.g., SQL, Cypher, etc.) by typing on the keyboard and executed via enter. In order to quickly find a table, they are listed on the menu for selection and navigation to the visual object.

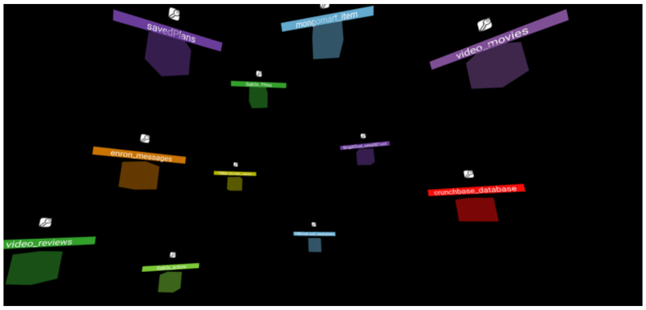

Figure 8. VRiDaM of MongoDB with dbkoda example data [26], showing spatial orientation and not intended to be readable.

Figure 9. VRiDaM of Neo4j with Northwind example data.

Figure 8 shows the VRiDaM visualization for MongoDB with dbkoda example data [26] (a Northwind port was no longer available), while Figure 9 shows Neo4j with Northwind example data.

V. EVALUATION

To evaluate VRiDaM, we compared its usage with a typical 2D database tool, DbVisualizer 10.0.13 (Free), shown in Figure 10 with the Northwind database loaded to provide an impression of the model's complexity, not meant to be readable. A convenience sample of eleven computer science students was selected. One experienced VR sickness symptoms and thus only the remaining ten were included in the results. All indicated they had some familiarity with SQL and they lacked NoSQL experience, so we chose to compare VRiDaM with an SQL tool. Three had not experienced VR before. The subjects were randomly selected to start in either VR mode (6) or the common tool (4). PostgreSQL Version 10 with the Northwind sample database (Figure 11Figure 10.) was used. Java 8 update 151, Apache Tomcat v8.0, AFRAME 0.8.2, Firefox 61, and SteamVR Version 2018-05-24 (1527117754).

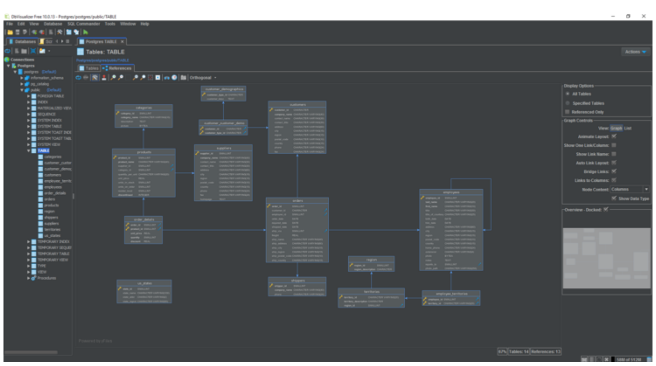

Figure 10. Northwind Traders data model in DbVisualizer.

These database tasks were given verbally and equivalent but not the exact same five questions asked in the other mode:

1) Which tables have a relation to table X?
2) To which table(s) does the table X have a relation?
3) What columns does table X have?
4) What are the foreign or primary keys of table X?
5) What are the keys in table X?

TABLE II. VRIDAM VS. DBVISUALIZER TASKS (AVERAGED)

	VRiDaM	DbVisualizer
Task duration (mm:ss)	4:48	1:46
Cumulative answers given (total/incorrect/missing)	130/6/4	140/1/6
Task correctness	92%	95%

The tasks results are shown in Table II. VRiDaM task duration was 2.7 times longer, and this can be expected since VR requires navigation time through space that 2D tools do not incur. The number of correct answers across the five tasks were 13 in VR and 14 in DbVisualizer, with ten subjects resulting in 130 or 140 cumulatively correct answers respectively. These longer VR task durations may be acceptable for certain user scenarios, and gives insight into what liabilities can be expected. A correctness difference of 3% across ten subjects is not necessarily significant and shows that the users were able to immerse themselves within minutes into a Benediktine space paradigm and perform tasks effectively. The task correctness differences could be attributed to personality, human alertness, distraction, or other factors beyond the paradigms or VR influence, as only 4 subjects in VR and only 3 subjects in non-VR were responsible for all errors, the rest had no mistakes.

Subjective impressions are shown in Figure 11 for VRiDaM intuitiveness and suitability of the controller interface and visualization as well as overall enjoyment. We note no significant difference between the interaction and the visualization intuitiveness or suitability. Only one preferred VRiDaM. This may indicate that more training and experience would be needed for them to feel more comfortable with a VR tool than with a 2D tool. Various debriefing comments indicated that the Benediktine spatial arrangement was either liked or not an issue for the subjects. When debriefed about what they liked about VRiDaM, comments included that it was a better database-model visualization, that tables were real objects instead of text boxes, how tables where displayed in space, and the highlighting effect.

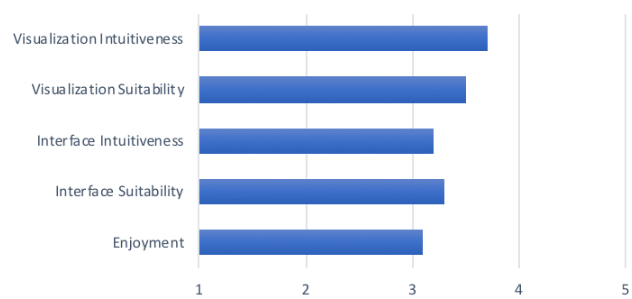

Figure 11. Average responses for VRiDaM (scale of 1 to 5 with 5 best).

 ISBN: 978-1-61208-668-2

The evaluation shows some of the challenges in utilizing VR for database-model visualization and interaction. VR object interaction is not standardized nor do users have familiarity and experience with it as they do with 2D mouse-based user interfaces. While VR enables new immersive paradigms and metaphors, these are not necessarily immediately intuitive. VR movement (moving the camera perspective) is more time consuming than scrolling or zooming a 2D perspective. For simpler tasks, VR tends to require more interaction time to accomplish the same task and thus entails efficiency costs. A 3D space permits objects to hide other objects, and for opaque objects requires movement to determine that no other objects are hidden. Given that the subjects were already familiar with E-R diagrams and SQL tools, yet had no prior training with VRiDaM and Benediktine space, we are satisfied with the ratings on suitability and intuitiveness. Based on comments about what subjects liked about VRiDaM we see it as a positive indicator and intend to investigate this approach further.

VI. Conclusion

This paper contributes VRiDaM, an immersive WebVR heterogeneous database visualization approach, applying Benediktine space visualization and force-directed graphs to (relational and non-relational) database models. It thus avoids the connection "yarn-balls" other techniques have in visualizing connections by leveraging spatial locality. A prototype was used to verify its viability and an empirical study evaluated its usability.

The empirical evaluation showed VRiDaM to be less efficient for equivalent analysis tasks while correctness was slightly worse. Intuitiveness, suitability, and enjoyment were given a better than neutral rating on average. One case of VR sickness occurred and we hope to address it in future work.

One ongoing challenge for a generic tool approach is the plethora of non-standardized interfaces to NoSQL and other databases. However, by providing driver plugins we believe that the adaptation overhead is small in relation to the advantages of a VR visualization that VRiDaM brings. Future work includes a more comprehensive empirical study and will investigate various optimizations to improve usability, performance, and scalability.

Acknowledgment

The author thanks Christoph Bauer and Camil Pogolski for assistance with the design, implementation, and evaluation.

References

[1] Cisco. The Zettabyte Era: Trends and Analysis. (Jun 7, 2017). Whitepaper. Available from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html 2018.07.24

[2] D. Reinsel, J. Gantz, and J. Rydning, "Data Age 2025: The Evolution of Data to Life-Critical," IDC Whitepaper (April, 2017).

[3] R. Cattell, "Scalable SQL and NoSQL data stores," Acm Sigmod Record 39, no. 4, pp. 2-27, 2011.

[4] P. Chen, "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned," In Software pioneers. Springer-Verlag, pp. 296-310, 2002.

[5] T. Merel, "The reality of VR/AR growth," Jan 11, 2017. Available from http://techcrunch.com/2017/01/11/the-reality-of-vrar-growth/ 2018.07.24

[6] M. Benedikt, "Cyberspace: some proposals," In Cyberspace, Michael Benedikt (Ed.). MIT Press, Cambridge, MA, pp. 119-224, 1991.

[7] P. Young, "Three Dimensional Information Visualisation," Technical Report, University of Durham, 1991.

[8] U. Fayyad, A. Wierse, and G. Grinstein (Eds.), Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann, 2002. ISBN: 1558606890.

[9] S. Fabrikant and B. Buttenfield, "Formalizing Semantic Spaces for Information Access," Annals of the Association of American Geographers, 91(2), pp. 263-280, 2001.

[10] D. Butler, J. Almond, R. Bergeron, K. Brodlie, and R. Haber, "Visualization reference models," In Proc. Visualization '93 Conf., IEEE CS Press, pp. 337-342, 1993.

[11] A. Bierbaum et al, "VR Juggler: A virtual platform for virtual reality application development," In Proc. IEEE Virtual Reality. IEEE, pp. 89-96, 2001.

[12] F. Fittkau, A. Krause, and W. Hasselbring. "Exploring software cities in virtual reality," In Proc. IEEE 3rd Working Conf. on Software Visualization (VISSOFT). IEEE, pp. 130-134, 2015.

[13] A. Teyseyre and M. Campo, "An overview of 3D software visualization," In IEEE Trans. on Visualization and Computer Graphics. IEEE, 15(1), pp. 87-105, 2009.

[14] A. Kashcha, "Software Galaxies," Available from http://github.com/anvaka/pm/ 2018.07.24

[15] J. Rilling and S. Mudur, "On the use of metaballs to visually map source code structures and analysis results onto 3d space," In Proc. 9th Work. Conf. on Reverse Engineering. IEEE, pp. 299-308, 2002.

[16] P. McIntosh, "X3D-UML: user-centered design, implementation and evaluation of 3D UML using X3D," Ph.D. dissertation, RMIT University, 2009.

[17] M. Krzywinski, "Schemaball: A New Spin on Database Visualization," In Dr. Dobb's: The World of Software Development,, 2004.

[18] E. Olshannikova, A. Ometov, Y. Koucheryavy, and T. Olsson, "Visualizing Big Data with augmented and virtual reality: challenges and research agenda," J. of Big Data, 2(22), 2015.

[19] I. Herman, G. Melancon, and M. Scott Marshall, "Graph visualization and navigation in information visualization: A survey," In IEEE Transactions on visualization and computer graphics, 6(1), pp. 24-43, 2000.

[20] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis, Graph drawing: algorithms for the visualization of graphs. Prentice Hall PTR, 1998.

[21] S. Card, "Information Visualization," In: Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications (Ed: Jacko, J.), Third Edition. CRC Press, pp. 544-545, 2012.

[22] S. Palmer, Vision Science. MIT Press, Cambridge (USA) 1999, ISBN 978-0262161831.

[23] D. Norman, The Design of Everyday Things: Revised and Expanded Edition. Hachette UK, 2013.

[24] A. Van der Heijden, "Perception for selection, selection for action, and action for perception," Visual Cognition, 3(4), pp. 357-361, 1996.

[25] S. Card, J. Mackinlay, and B. Schneiderman (editors), Readings in Information Visualization: Using Vision to Think. Morgan Kaufman, 1999.

[26] [Online] Available from https://github.com/SouthbankSoftware/dbkoda-data 2018.08.12