

VR-EAT: Visualization of Enterprise Architecture Tool Diagrams in Virtual Reality

Roy Oberhauser¹[0000-0002-7606-8226], Pedro Sousa², and Florian Michel³

¹Computer Science Dept., Aalen University, Aalen, Germany
roy.oberhauser@hs-aalen.de

²Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal
Link Consulting, Lisbon, Portugal
pedro.manuel.sousa@tecnico.ulisboa.pt

³Computer Science Dept., Aalen University, Aalen, Germany
florian.michel@studmail.hs-aalen.de

Abstract. The digital transformation occurring in enterprises results in an increasingly dynamic and complex IT landscape that in turn impacts enterprise architecture (EA) and its artefacts. New approaches for dealing with more complex and dynamic models and conveying EA structural and relational insights are needed. As EA tools attempt to address these challenges, virtual reality (VR) can potentially enhance EA tool capabilities and user insight but further investigation is needed in how this can be achieved. This paper contributes a VR solution concept for visualizing, navigating, and interacting with EA tool dynamically-generated diagrams and models using the EA tool Atlas. An implementation shows its feasibility and a case study using EA scenarios is used to demonstrate its potential.

Keywords: Virtual Reality, Enterprise Architecture, Enterprise Modeling, Enterprise Architecture Tools, Visualization.

1 Introduction

A major *digital transformation* of industries is underway (Muro et al. 2017). While the digitalization rate (digital score) may vary across industries and economies, it is nevertheless impacting business strategies and necessarily the enterprise architecture (EA) that supports the business. Increased digital automation in turn affects business functions and processes, and thus affects and changes the actual EA. The IT infrastructure expands and becomes more dynamic and complex to support these both rapidly changing and highly-integrated business processes, resulting in a complex EA.

Enterprise architecture (EA) comprises the structural and behavioral aspects needed for an enterprise to function and adapt in alignment with some vision. To this end, it involves comprehensive and cohesive modeling and documentation. Considering the trends and challenges mentioned, the reality that EA is attempting to comprehensively model, document, and change has become much more complex than in previous decades. EA seeks to provide a comprehensive set of cohesive models to describe

the enterprise structure and functions, while individual models are logically arranged to provide further detail about an enterprise (Jarvis 2003). Healthy enterprises necessarily evolve, and they thus need to maintain explicit knowledge of their EA, be it for enterprise governance, engineering, compliance, maintenance, etc. Architectural representations are an enterprise asset that must be governed (Hoogervorst 2009). However, the effort required to keep architectural views updated is known to be very high in current organizations (Sousa 2018), mainly because the organization's structure is the result of an asynchronous, distributed, and heterogeneous process, producing representations in different languages/notations, with different levels of detail, in different tools at different times. But timepoints are actually a more complex issue, because enterprise models are also a moving target, as enterprises need architectural views that refer to different points in time, namely past, present and future. So, one aspect of our research has been in finding a low effort method that supports updated architectural views regardless of the point in time (past, present or future).

The Atlas tool (Sousa 2018) was developed under the Enterprise Cartography paradigm (Sousa 2009)(Tribolet 2014) to minimize the effort required to keep architectural views up-to-date in fast changing organizations. In Atlas, all architecture views are generated on-the-fly and include a time bar where one can navigate in time and see the architecture evolution from any point in the past to any point in the future. The view's contents regarding the future are computed, processing the plans of transformation initiatives pipeline (both ongoing and planned) to produce a consolidated state in any point in time. Therefore, one can foresee the contents of an architecture view in some desired future date by consolidating the current view's content with the expected changes of ongoing and planned transformation initiatives whose completion date precedes the desired date (Sousa 2009)(Sousa 2018a). Atlas is also able to generate and support time navigation in BPMN models (Sousa 2019)(Colaço 2017)(Cardoso 2020).

In general, modeling provides an abstracted or simplified representation of a system that can assist with understanding relationships between elements or concepts of interest. Typically, views are used to address stakeholder concerns and portray relevant aspects of a model. However, with typical 2D view depictions, one can lose insight into the interrelationships across views and relevant model elements. For certain EA-related tasks, as EA complexity increases, the need to visualize and inspect multiple related model views is limited by the current 2D capabilities.

Virtual Reality (VR) could potentially assist with visualizing this growing and complex set of models and their interrelationships simultaneously in a spatial structure. VR is defined as a “real or simulated environment in which the perceiver experiences telepresence” (Steuer 1992), a mediated visual environment which is created and then experienced. VR has made inroads in various domains and become readily accessible as hardware prices have dropped and capabilities improved. As EA models grow in complexity and reflect the deeper integration of both the business and IT reality, an immersive EA environment could provide an additional visualization capability to comprehend the “big picture” for structurally and hierarchically complex and interconnected diagrams, while providing an immersive experience for EA models in a 3D space viewable from different perspectives.

In prior work, VR-EA (Oberhauser & Pogolski 2019) presented a VR solution concept for visualizing, navigating, annotating, and interacting with ArchiMate (Open Group 2017) EA models and our generalized VR modeling framework (VR-MF), while VR-BPMN (Oberhauser et al. 2018) described a solution concept for visualizing Business Process Model and Notation (BPMN) (OMG 2011) models in VR. The Atlas EA tool and meta-model (Sousa 2018) consolidates various EA model and data sources into a single repository and provides coherent visualization and view capabilities. This paper presents our solution concept VR-EAT (EA Tool) for integrating EA tool visualizations into VR, in particular making any number of dynamically generated EA diagrams from the EA tool Atlas available and enhancing these with 3D depth, inter-diagram element connections. By leveraging the unlimited space in VR, overall interrelationships of the models and views can be indicated and considered adjacent to one another, while sensory immersion can support task focus while limiting visual distractions that typical 2D display surroundings incur. The remainder of this paper is structured as follows: Section 2 discusses related work while Section 3 provides background detail on the EA tool Atlas. In Section 4 our solution concept VR-EAT is described. Section 5 then provides details on our prototype implementation. The evaluation is described in Section 6, and a conclusion follows in Section 7.

2 Related Work

Work related to EA visualization includes (Rehring et al. 2019), who applied 3D visualization in augmented reality in support of EA decision making. (Naranjo et al. 2014) describe PRIMate based on PRIMROSe, a visual graph-based enterprise analysis framework, and show a graph, treemap, and 3D visualization of an the ArchiSurance ArchiMate model. As to harmonizing ArchiMate, BPMN, and UML, (van den Berg 2012) analyzes the various metamodels and shows how one could practically combine the notations across views and diagrams. We are unaware of research applying VR to the EA area with EA tool (meta-)model and non-standard multidiagram integration into VR, including support for model visualization the EA standards ArchiMate and BPMN to VR.

Our VR-EAT solution concept enables VR-centric visualization with integration with the EA tool Atlas. It is implemented with standard game engine technology (Unity) using common VR hardware. It supports hypermodeling, e.g., combining ArchiMate, BPMN, and EA tool models in the same space, provides automatic layout of views as stacked 3D hyperplanes, visualizes the reality of inter-view relations of elements, and integrates non-standard EA diagrams for an immersive experience to support deeper EA analysis across multiple diagrams and stakeholder concerns.

3 Background on the EA Tool Atlas

Atlas is an EA tool with a repository holding a fully configurable metamodel that generates views (also fully configurable), including all the information required to represent the views at any point in time as well as representing each artifact in its

lifecycle state (Sousa 2018). Thus, architectural views are movies, not pictures. This is a unique feature of Atlas and, in our experience, a fundamental one to reduce the effort of maintaining architectural views in large organizations.

Atlas is far from providing all the architectural view types one may find in well-known tools (Roth 2014), but it supports the time-travel mechanism in all the views types it supports. It supports the configuration of the view types according to the metamodel defined by the user. Fig. 1 shows some view types supported by Atlas.

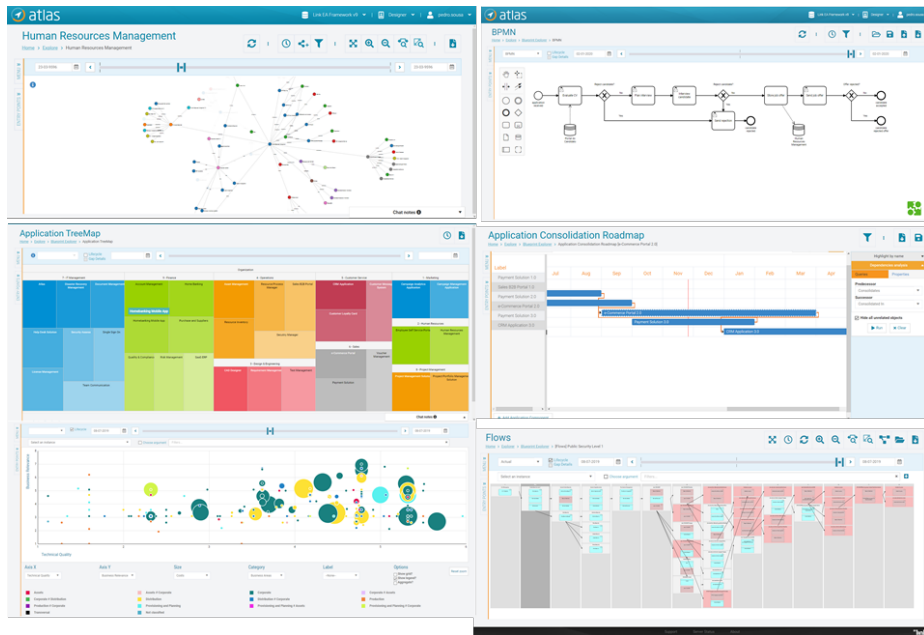


Fig. 1. Various view types supported by Atlas.

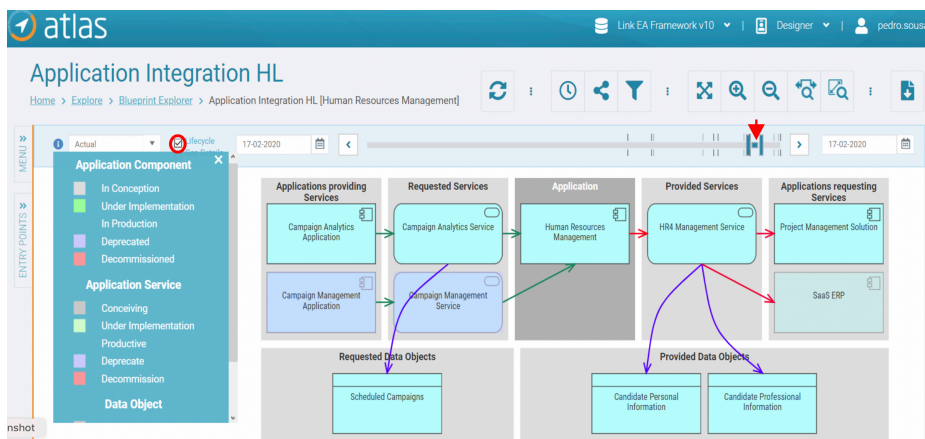


Fig. 2. High-level Application Integration Blueprint for HR Management application.

A key aspect Atlas addresses is the generation of up-to-date architectural views with near zero effort as has been done for some cases (Sousa 2011)(Sousa 2014), even with the previous generation of the tool named EAMS. Users define templates of architectural views that are instantiated on request to particular objects. Fig. 2 illustrates the generated view for the Human Resources (HR) Management application, presented in the middle container. Moving leftwards, one can see the Services requested by the HR Management application and the applications providing such services. Moving rightwards, one can see the service realized by the HR Management application and the applications that request it. Below it data objects used in the services.

Since the Lifecycle option is selected (red circle), artefacts symbols are shaded according with their lifetime state on the date defined by the time handler position in the time bar (top red arrow). The legend on the left presents the lifecycle states defined for Application Components and Application Services. In this case, on the selected date (17/02/2020) the Campaign Management application and service are Deprecated (light blue on the left), and the SaaS ERP application is In Conception (grey, on the right). To feed Atlas with the information required to produce such architectural views, one uses the information made available by project plans, be it simple lists of created and decommissioned artefacts or models in some notation such as ArchiMate. In this last case, since ArchiMate does not provide a way to state that some work package creates/deletes/or changes any artefact, one uses association relationships named as "created by", "decommissioned by" or "changed by". Atlas also provides a transformation engine that allows end users to configure how each concept in an imported model (such as in ArchiMate) maps into the concepts defined in its metamodel.

Atlas also allows end users to define the propagation rules between a project milestone dates and the lifecycles of the artefacts created/decommissioned and changed by the project. A default rule for artefact creation is that objects created by a project become productive upon project completion. So, whenever a project is delayed, Atlas can update the lifecycle of dependent artefacts. Finally, end users can also define the rules the establish dependency between projects. A default rule is that a project A is dependent of a Project B if it uses some artefact created, changed or decommissioned by project B. So, whenever a project termination date is delayed, Atlas can alert (e.g., email) the actor responsible for the impacted ongoing or planned projects.

4 Solution Concept

As shown in Fig. 3, our generalized VR Modeling Framework (VR-MF) (Oberhauser & Pogolski 2019) provides a VR-based domain-independent hypermodeling framework that addresses three primary aspects of modeling in VR: visualization, navigation, and interaction. Rather than requiring unique and specific 3D shapes to represent various model elements, since most models already have 2D graphical element notations, we chose the more pragmatic principle of utilizing cubes to represent elements, since its sides can be viewed from all perspectives an optional 2D type icon can easily be portrayed as a material on the cube's sides. Furthermore, relationships between

elements can be shown in 3D space, and related elements can be grouped in 3D layers or views as appropriate. VR-EA provides specialized direct support and mapping for EA models in VR (Oberhauser & Pogolski 2019), including both ArchiMate as well as BPMN (VR-BPMN) models (Oberhauser et al. 2018). VR-EAT (EA Tool) is our solution concept for integrating EA tool data and visualizations in VR. The VR-EAT solution concept is generic and its feasibility is demonstrated with the EA tool Atlas.

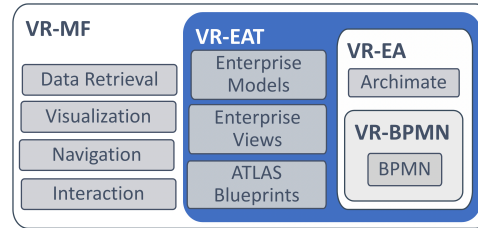


Fig. 3. The VR-MF (general) and VR-EAT (EA Tool) solution concept

ArchiMate models use a graphical notation consisting of a collection of concepts to portray a wide scope of EA elements and relationships. Elements can be behavioral, structural, motivational, or some composite. These concepts can participate in various layers: strategy, business, application, technology, physical, and implementation & migration, the layers having colors associated with them. Cross-cutting aspects involved include: passive structure, active structure, behavior, and motivation. Views are used to convey information addressing concerns of specific stakeholders. On the other hand, BPMN models focus on business processes and consist of Business Process Diagrams (BPDs) composed of graphical elements consisting of flow objects, connecting objects, swim lanes, and artifacts (OMG 2011). On the other hand, Atlas, as a representative EA tool, provides access to diverse EA-related data in a coherent repository and meta-model and is not restricted to certain standards or notations.

Visualization. While many visual options and metaphors are possible, diverging too far from the 2D diagrams and notations familiar to EA tool users would reduce comprehension. Yet to place 2D images like flat screens in front of users would provide little added value. As a transitional hybrid before we have full VR EA tooling, to differentiate elements by type 3D boxes provide depth in the diagrams and project the relevant 2D object type notation from the EA tool onto all sides as a texture, which can thus be perceived from all angles. In contrast to 2D space, one challenge in 3D space element placement is that one can never be sure if an element is not hidden behind another element at any particular vantage point if the element is opaque. However, if one makes the element partially transparent, then it can become confusing as to which element one is actually focusing on. We thus chose to make the elements opaque in order to avoid this visual confusion, and by briefly adjusting one's perspective one can visually check that nothing is hidden behind an element. Moreover, visualizing text is an issue in VR due to the relatively low resolutions currently available and the distance from the virtual camera position to the text. Also, labels for elements can differ widely in length, yet should not interfere with understanding the underlying

diagram structure. We thus place element labels on the top of the sides. For dealing with longer element labels, on selection (when element is of interest) the full label is displayed. For visualizing and differentiating the various diagrams, hyperplanes are used to take advantage of the 3D space, with each plane representing one diagram.

Navigation. The immersion afforded by VR requires addressing how to intuitively navigate the space while reducing the likelihood of potential VR sickness symptoms. Two navigation modes are included in the solution concept: the default uses gliding controls, enabling users to *fly through* the VR space and get an overview of the entire model from any angle they wish. Alternatively, teleporting permits a user to select a destination and be instantly placed there (i.e., by moving the camera to that position); this can be disconcerting but may reduce the likelihood of VR sickness that can occur when moving through a virtual space for those prone to it.

Interaction. VR interaction with VR elements has not yet become standardized. In our VR concept, user-element interaction is done primarily via the VR controllers. Views are stacked hyperplanes and can be made visible or invisible by selecting the plane or equivalent icon. Inter-view connections can be enabled or disabled. A specific connection can be selected by selecting an element to emphasize it. To reduce visual clutter, one mode permits visualizing only the inter-view connections for the element of interest, hiding all others.

5 Realization

Fig. 4 shows our solution concept realization for VR-EAT. The Unity block denotes our VR-MF realization, and besides Atlas Blueprint views includes direct support for ArchiMate and BPMN models (VR-EA). A Data Hub is shown in the center while below it, MongoDB is used for local data storage and to update data within Unity. The top left shows the integration with the EA tool Atlas, including repository data and service access via REST/JSON. A command line extension shown on the left provides helper functions for configuration, mapping, and data loading for the Data Hub. Initially, due to time and resource constraints, our realization primarily addresses visualization and navigation of existing views and models in Atlas. Future work will include support for creating and improving views and models from within VR.

The integration with Atlas was done in a VirtualBox VM with Win10Pro 1903 with 8GB RAM, 4-core CPU @4.6GHz, 50GB. The VR-EAT implementation used Unity 2019.1.14f1 with IL2CPP, Blender 2.8, SteamVR 1.9.16, MongoDB 4.2.2. The Data Hub uses NuGet packages CommandLineParser 2.7.82, MongoDB.Driver 2.10.0, Newtonsoft.Json 12.0.3, and .NET system packages. The Data Hub, Command Line Extension, and Unity run on the .NET Framework 4.6.

Our sample Atlas repository contained about 66 core blueprints without parameter choices, and resulting in 7900 different blueprints considering all selection combinations. We load these in the Data Hub and save them to MongoDB in our schema, to permit us to transform and annotate the data as needed for VR.

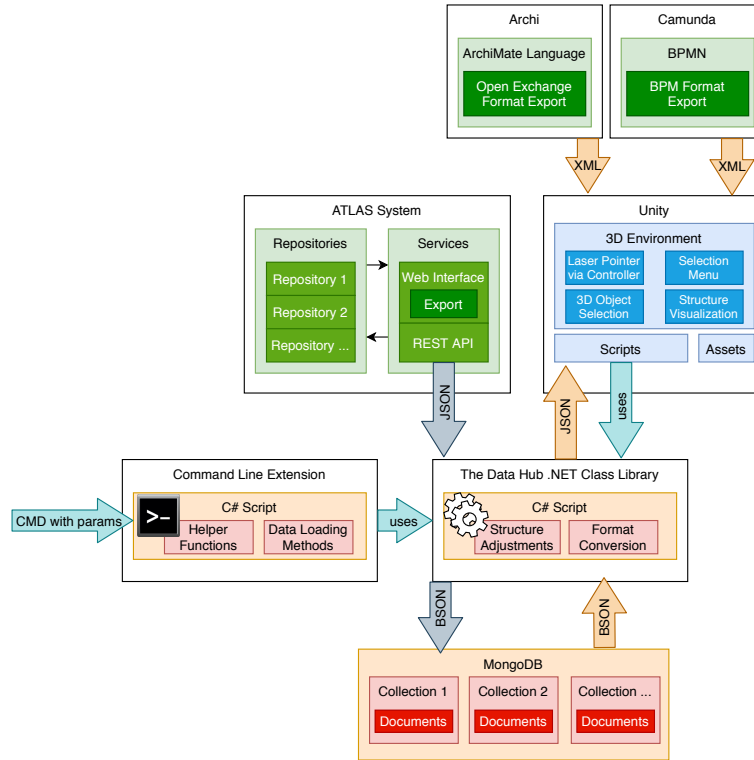


Fig. 4. VR-EAT solution concept realization.

Initially an empty blueprint placeholder (Fig. 5 top) is shown with menu options as a local menu attached to the plane. After menu option parameters have been specified, the generate button (checkmark) causes the VR blueprint to be displayed (Fig. 5 bottom) equivalent to that of Atlas (Fig. 6). In VR, the elements in red are elements that do not exist at that timepoint, having been decommissioned (past) or to exist (future). The top diagram also shows an empty placeholder after adding a view (+).

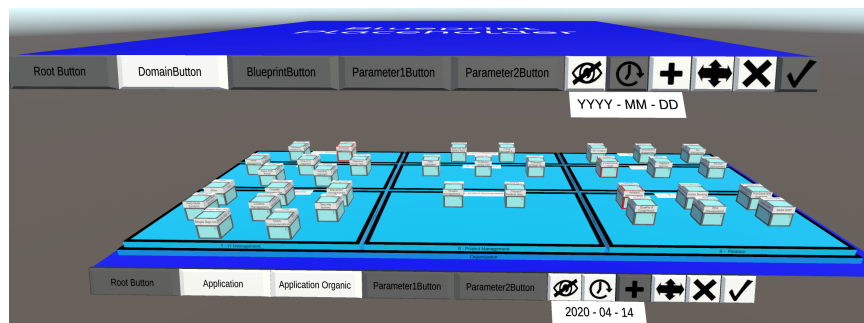


Fig. 5. Application / Application Organic blueprint in VR.

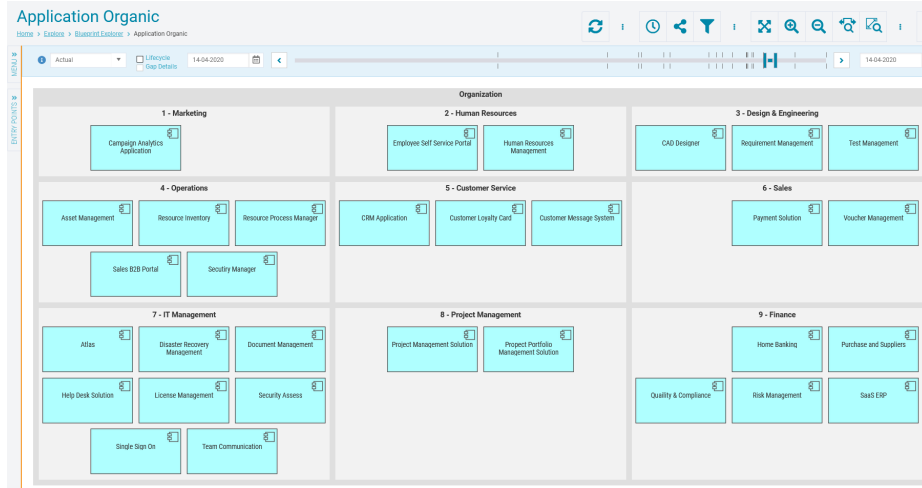


Fig. 6. Application / Application Organic blueprint in Atlas.

Our menu options concept (Fig. 8 right) includes a floating Global Options Menu (realized as Fig. 7 left) and a local Options Menu (realized as Fig. 7 right). As to our solution concept for organizing spatial placement of Atlas blueprints, a stack-based scheme (Fig. 8 left) with a maximum of three parallel stacks is used (arbitrary to consolidate diagrams in relative proximity), with no limit on the views in any stack. The floating Global Options Menu supports deleting all views, setting a timepoint for all views, and saving/loading the configuration, while the local blueprint Options Menu supports Domain/Blueprint/Parameter, hide (eye symbol), timepoint (Fig. 7 center), add stack element above (+ symbol), move element within/between stacks (arrows in Fig. 7 right), delete element (x symbol), generate blueprint based on parameter selection (checkmark), and domain selection (leftmost Root button) to switch between root repositories.

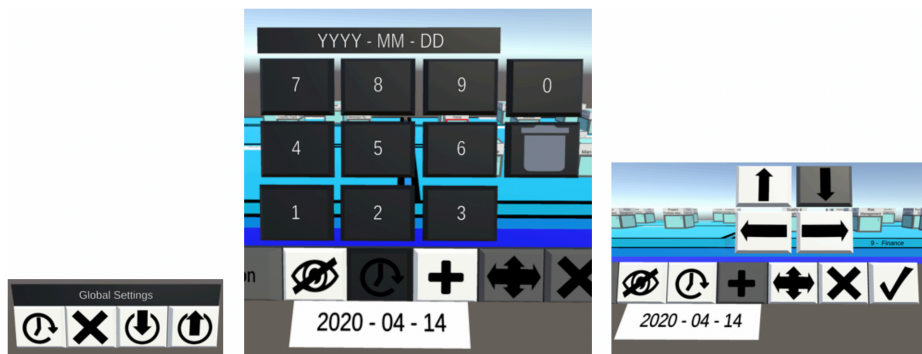


Fig. 7. VR-EAT menus: floating global (left), timepoint (center), and local (right).

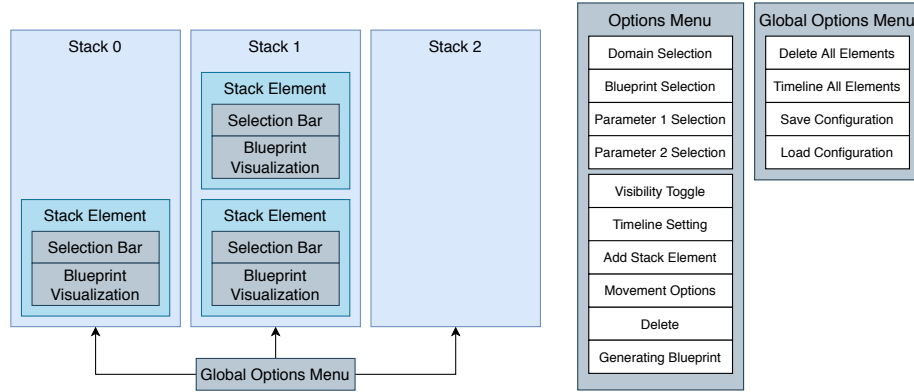


Fig. 8. VR-EAT triple stack concept (left) and local and global menu selection options (right).

Since hundreds of parameters are possible, selection of the domain/blueprint/parameter1/parameter2 button dynamically shows the available choices as buttons upon selection (see Fig. 9 concept and Fig. 10 VR implementation).

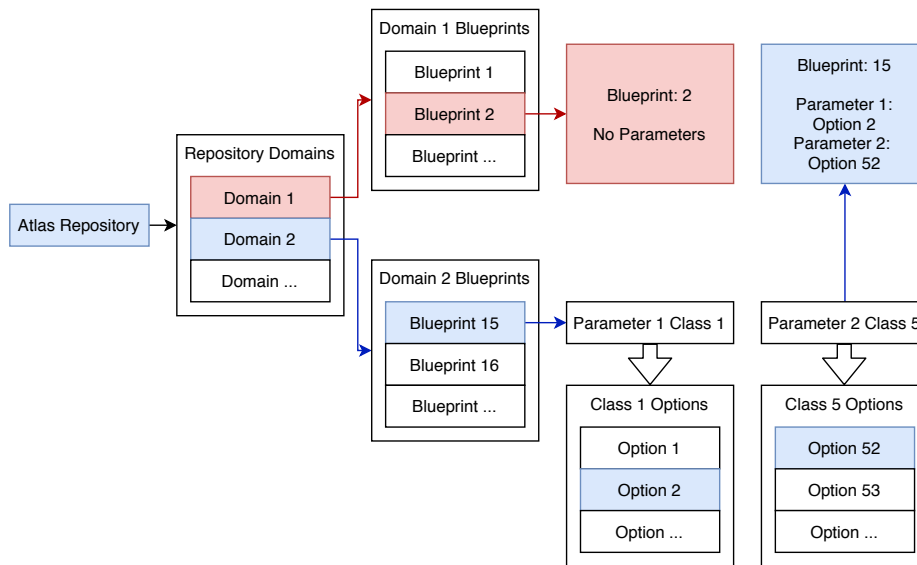


Fig. 9. VR-EAT local blueprint selection.

Repository	Risk	System Software		
Goal	Information	ATLAS	Architecture	Node
Business Process	Business Service	Data Object	Database Schema	Flow
Project	Business Actor	Business Function Flows	Business	Business Object
Application	Application Data	Application Release	Application Service	Unnamed
Root Button		DomainButton	BlueprintButton	Parameter1Button

Fig. 10. VR-EAT local blueprint selection.

Fig. 11 shows elements with icons and labels placed on the sides, containment is indicated with raised, colored, and named plates, dependencies are shown with black connectors, and a selected element is indicated with a pink glow and an indicator of that same element in other views.

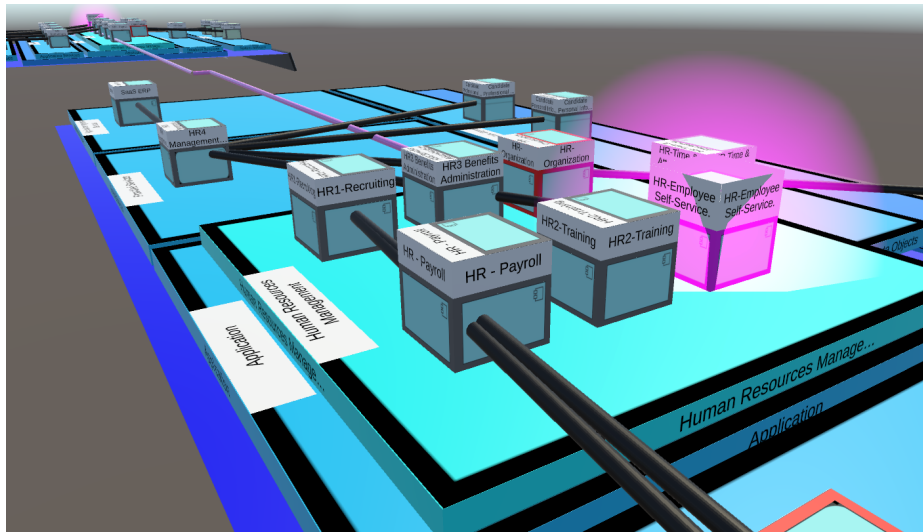


Fig. 11. VR-EAT container and element selection (glow) visualization.

In Fig. 12, a side-by-side comparison of timepoints 1-1-2010 on left and 16-03-2020 on right is shown, whereby in Atlas two browser windows are shown (top) and in VR-EAT (bottom) a side-by-side depiction is supported. Fig. 13 shows support for multiple EA models side-by-side in VR to support analysis, with VR-EAT Atlas blueprints (left) and (on the right) VR-EA with the ArchiSurance Archimate model.

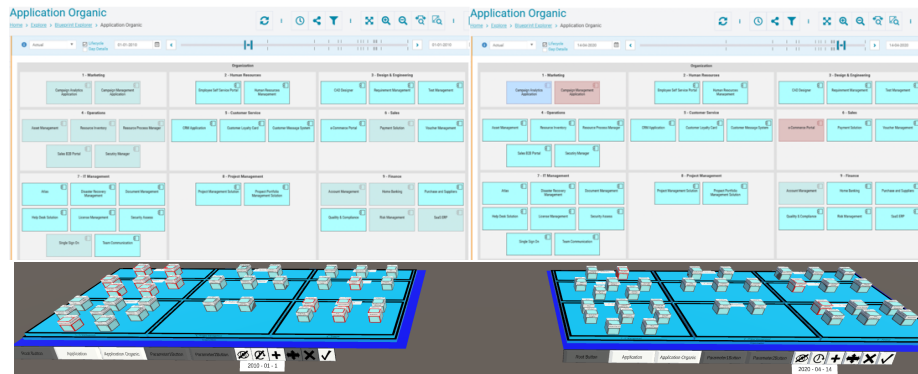


Fig. 12. Time delta in Atlas (top) in 2 browser windows with 1-1-2010 left and 14-04-2020 right, and equivalent VR-EAT with via visual marking in red of non-current elements (bottom).

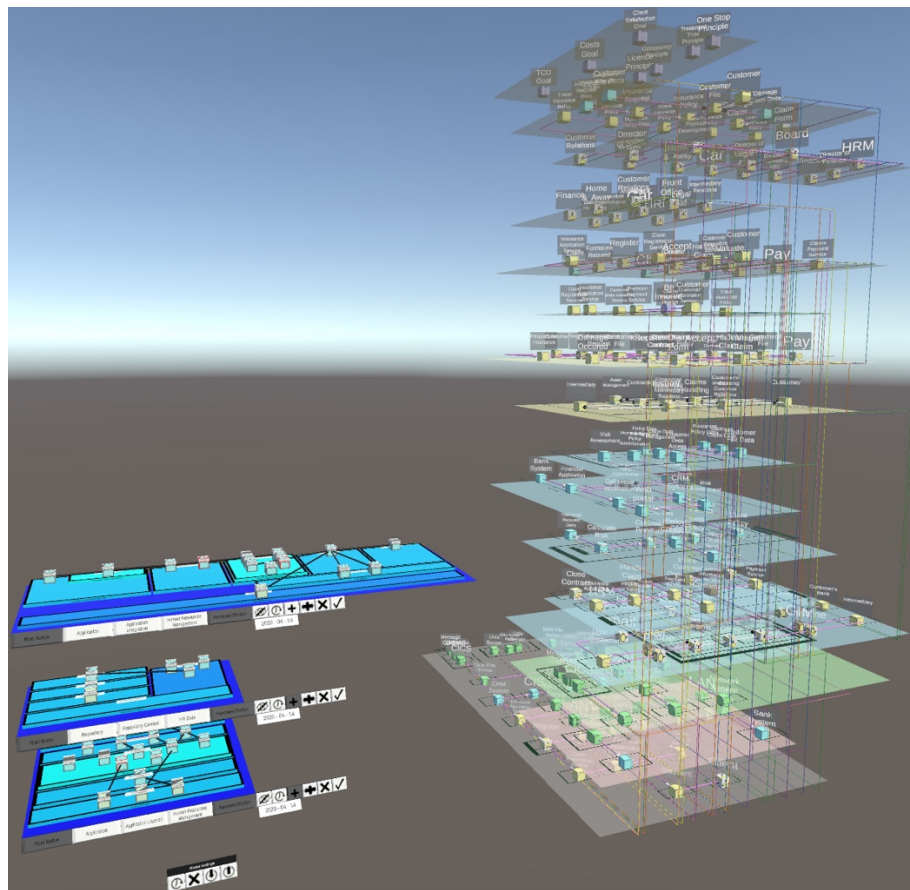


Fig. 13. VR-EAT showing Atlas blueprints (left) and VR-EA with the ArchiSurance Archimate model (right) side-by-side in VR to support analysis.

6 Evaluation

A case study is used to evaluate the practicality of the VR-EAT concept and implementation. Two scenarios that require multiple views (blueprints) to assess the situation will be used to exemplify usage of VR-EAT: (1) EA system replacement and (2) data storage consolidation.

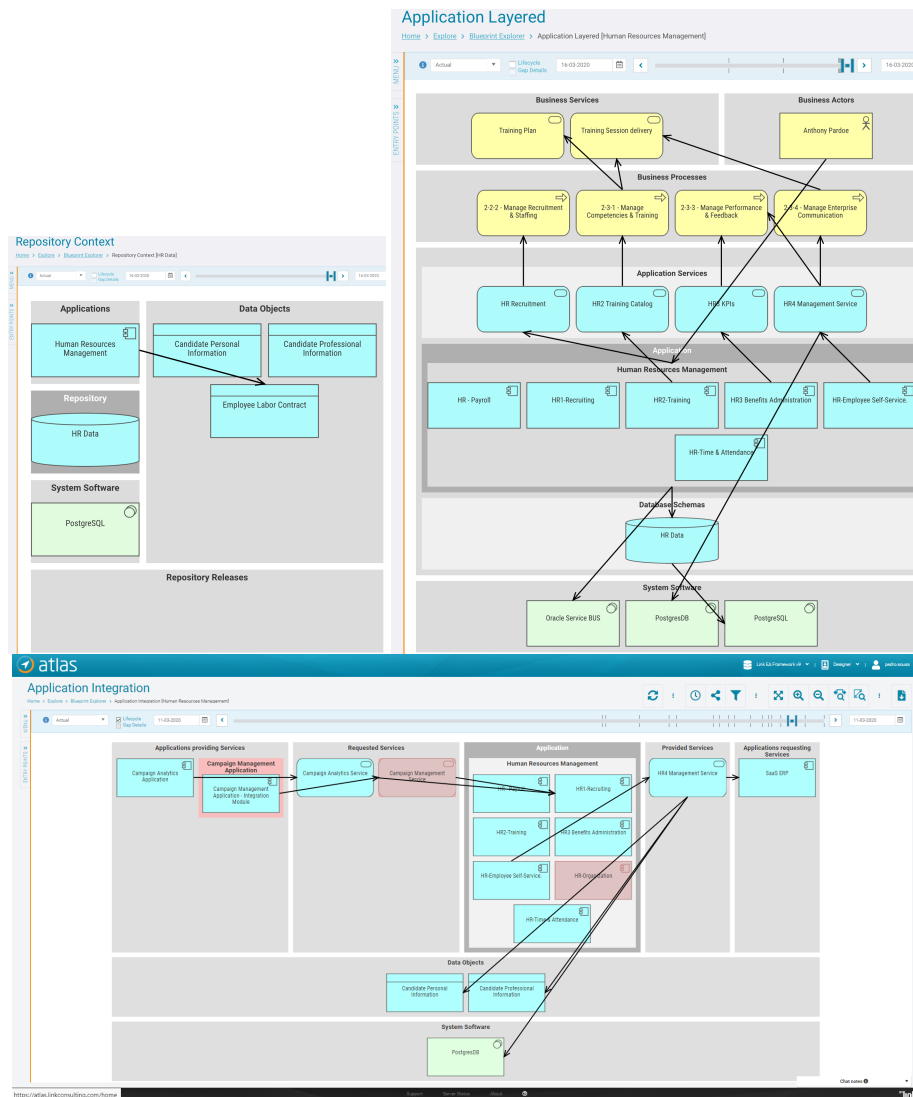


Fig. 14. Repository Context [HR Data] (top left), Application Layered [HRM] (top right), and Application Integration [HRM] (bottom) blueprints in Atlas.

6.1 System Replacement Scenario

When considering system replacement (perhaps due to increased licensing costs), various blueprints must be consulted to identify dependencies of the system, both current and planned at the time of system replacement. First with the application landscape blueprint is used to drill down into the Application Layered [Human Resource Management] (HRM) blueprint (Fig. 14 right) and determine any impacted business processes: one sees four application services provided by this application and the business processes where they are used. Second, the same blueprint shows the underlying system software used: an HR database running on Postgres and the Oracle Service Bus. Third, the Repository Context [HR Data] blueprint (Fig. 14 left) is used to check what else uses the HR Data, and the business objects in that database. Finally, the Application Integration [HRM] blueprint (Fig. 14 bottom) is used to determine any integrations with other systems, where one can see that only one system is consuming application services from HRM. In VR-EAT, the equivalent is shown in a side-by-side (Fig. 15) and stacked (Fig. 16) configuration with the element of interest (HRM system) selected.

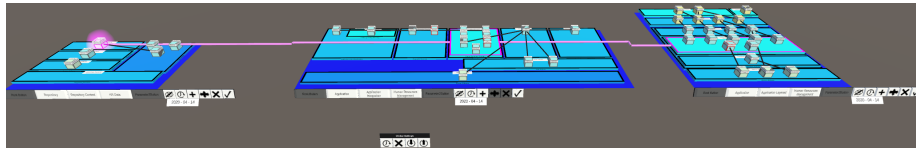


Fig. 15. VR-EAT in side-by-side configuration of three HRM system replacement blueprints with the HRM system selected.

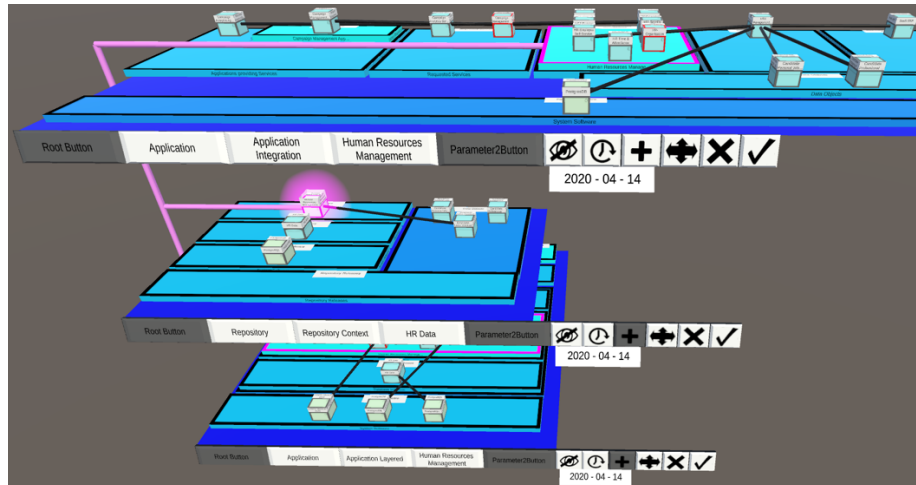


Fig. 16. The three HRM Application system replacement blueprints stacked in VR-EAT.

6.2 Data Storage Consolidation Scenario

Views are specific to a stakeholder and their interest. In larger enterprises with many possible blueprints (our sample database alone offers 7900), the question that can linger in the mind of an EA analyst is: Did I miss something? Have I looked at all the relevant diagrams? Have I considered all possible impacts?

For this case study, let us consider the question if consolidation of databases onto one product/vendor is realistic? How many different ones are even used, and by which applications for what scenarios? How critical are these to the business?

For that, multiple blueprints might be used. We suggest that a hybrid approach of non-VR and VR be used. Direct use of Atlas (non-VR) would be more efficient for quickly ascertaining the various relevant blueprints, for example the four blueprints as shown in Fig. 17 where Application Layered [Financial] (left) and Repository Context [HR Data] (right) both involve PostgreSQL and Fig. 18 where System Software Layered [PostgreSQL] (left) shows applications using PostgreSQL and Application Context [Financial] (right) involves use of the Oracle Service Bus and may affect vendor lock-in. Once these are identified, in VR-EAT these could be stacked or laid side-by-side for deeper analysis and element of interest selection in VR as shown in Fig. 19.

6.3 Discussion

As to the benefits of an immersive VR experience vs. 2D, (Müller et al. 2014) investigated a software analysis task 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 mistakes were made). While interaction time was less efficient, VR improved the user experience (more motivating, less demanding, more inventive/innovative, and more clearly structured). Analogously, we believe VR-EAT shows that EA tool dynamically-generated diagrams can be brought into VR to support EA stakeholders as EA complexity grows. VR-EAT benefits include: unlimited simultaneous diagrams without screen space limitations, structured placement of diagrams, inter-diagram dependencies. By providing an immersive VR experience to fly-in, navigate and get a feel for the context and surrounding elements involved, it removes other visual distractions in our surroundings that 2D displays inherently involve. It is analogous to being outside an aquarium versus swimming with the fish.

Thus, VR-EAT can support EA decision-making by making multiple EA views and models and their interdependencies simultaneously available, and in combination with VR-EA, these can be visualized and tied to other available EA models such as Archimate. VR-EAT can be positioned as providing a (transitional / hybrid) way to access and utilize the 2D diagrams and models that are common to EA users and available in EA tools today, and enable VR-based EA modeling to utilize these. VR does involve challenges, including limiting cognitive overload and visual clutter and ascertaining the intentions of the user, and as VR is still relatively new and not commonplace for IT work environments, interaction models with elements have not yet been standardized, incurring additional interaction and training overhead.

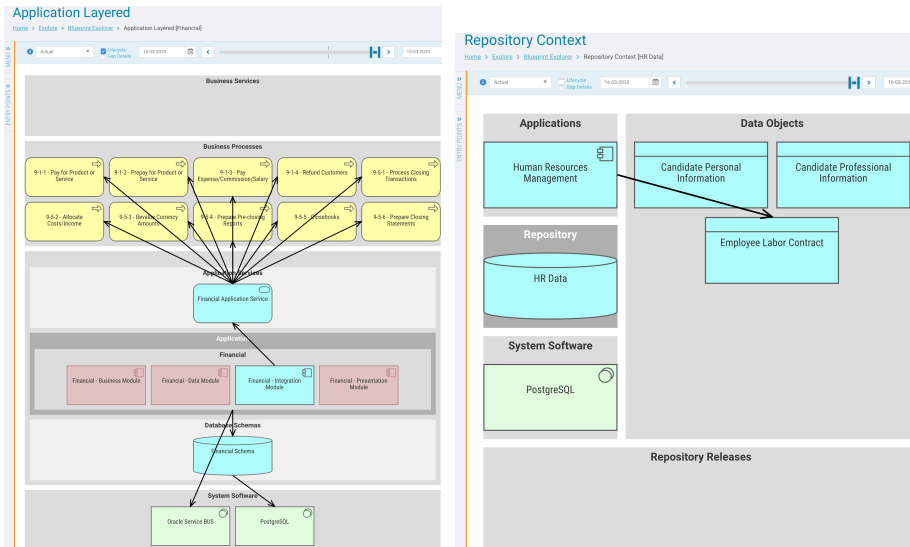


Fig. 17. Application Layered [Financial] (left) and Repository Context [HR Data] (right).

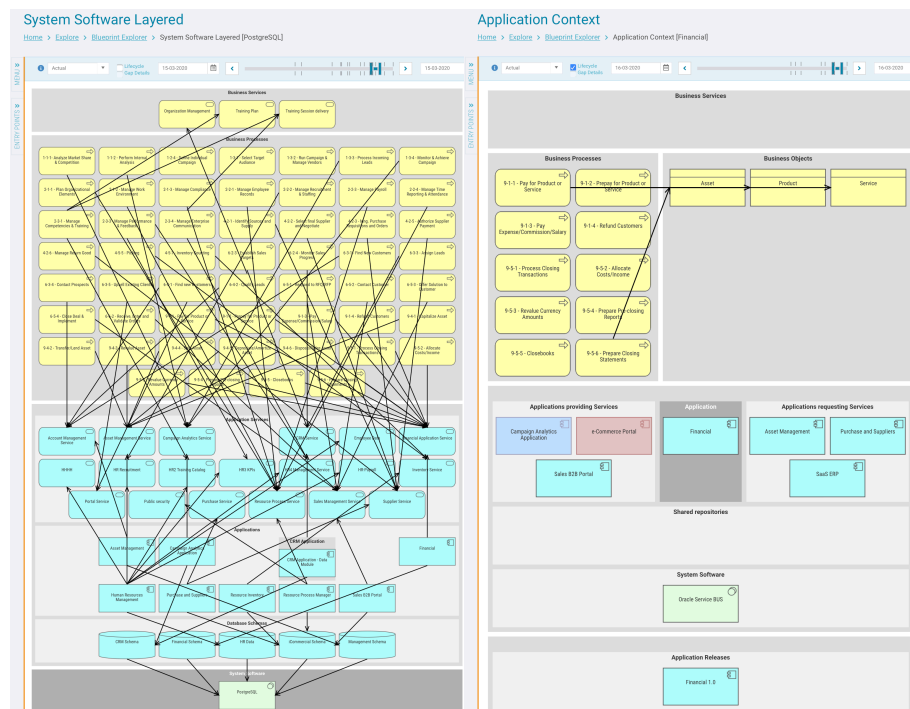


Fig. 18. System Software Layered [PostgreSQL] (left) and Application Context [Financial] (right).

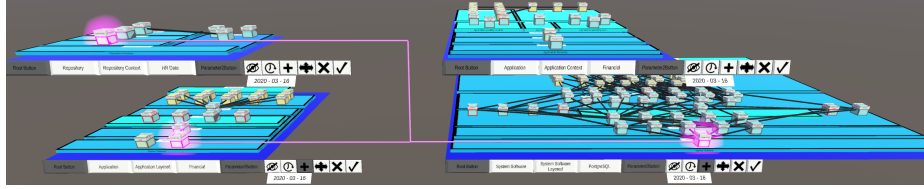


Fig. 19. Various database-related blueprints with selected element visualization in VR-EAT.

7 Conclusion

As enterprises evolve and EA models increase in complexity, new and practical approaches for integrating available EA tool diagram and model information sources and dynamically-generated diagram visualizations are needed. With VR-EAT we demonstrated the specific capability to integrate advanced EA tools such as Atlas and convey their available EA-related dynamic diagrams into VR for an immersive VR experience. VR-EAT is founded on the general VR modeling framework VR-MF and enhances VR-EA with direct EA tool support. A prototype implementation of VR-EAT demonstrated its feasibility with Atlas. Our evaluation case study showed its potential to support various EA scenarios that require analysis of multiple diagrams in conjunction with one another.

As EA complexity increases, we see a great potential in VR to visualize objects (that have symbolic meaning to the users) in a spatial format and allow them to navigate and interact with them in this immersive space. While views have their place for simplifying the enterprise reality for specific stakeholder interests, for more complex analytical tasks involving multiple stakeholders, 2D graphical visualizations are limited in the ability and area on which multiple diagrams and inter-model aspects are analyzed. Future work will enhance the interactive and informational capabilities of VR-EAT, provide a more integrated synchronization and immersive experience between the Atlas diagrams and other EA models, and investigate VR-EAT usage in an empirical study.

Acknowledgements. The authors would like to thank Ricardo Santos Leal for his technical assistance with Atlas.

References

1. Cardoso D., Sousa P.: Generation of Stakeholder-Specific BPMN Models. In: Advances in Enterprise Engineering Working Conference (EEWC 2019). LNBP, vol 374, pp. 15-32. Springer, Cham (2019)
2. Colaço, J., Sousa, P.: View Integration of Business Process Models. In: European, Mediterranean, and Middle Eastern Conference on Information Systems, pp. 619-632. Springer, Cham (2017).
3. Hoogervorst J.: Enterprise governance and enterprise engineering. Springer (2009).

4. Jarvis, B.: Enterprise Architecture: Understanding the Bigger Picture - A Best Practice Guide for Decision Makers in IT. The UK National Computing Centre (2003)
5. Müller, R., Kovacs, P., Schilbach, J., Zeckzer, D.: How to master challenges in experimental evaluation of 2D versus 3D software visualizations. In: 2014 IEEE VIS International Workshop on 3DVis (3DVis), pp. 33-36. IEEE (2014).
6. Muro, M., Liu, S., Whiton, J., Kulkarni, S.: Digitalization and the American Workforce. Brookings Institution Metropolitan Policy Program (2017). https://www.brookings.edu/wp-content/uploads/2017/11/mpp_2017nov15_digitalization_full_report.pdf
7. Naranjo, D., Sánchez, M., Villalobos, J.: Towards a unified and modular approach for visual analysis of enterprise models. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, pp. 77-86. IEEE (2014).
8. Oberhauser R., Pogolski C.: VR-EA: Virtual Reality Visualization of Enterprise Architecture Models with ArchiMate and BPMN. In: Shishkov B. (eds) Business Modeling and Software Design (BMSD 2019). LNBIP, vol 356. Springer, Cham (2019).
9. Oberhauser, R., Pogolski, C., Matic, A.: VR-BPMN: Visualizing BPMN Models in Virtual Reality. In: Shishkov B. (eds) International Symposium on Business Modeling and Software Design (BMSD 2018), pp. 83-97. Springer, Cham (2018).
10. OMG: Business Process Model and Notation (BPMN) Version 2.0. OMG (2011).
11. Open Group: ArchiMate 3.0.1 Specification. The Open Group (2017).
12. Rehring, K., Greulich, M., Bredenfeld, L., Ahlemann, F.: Let's Get in Touch - Decision Making about Enterprise Architecture Using 3D Visualization in Augmented Reality. In: Proc. 52nd Hawaii Int'l. Conf. on System Sciences (HICSS), pp. 1769-1778. IEEE (2019).
13. Roth, S., Zec, M., Matthes, F.: Enterprise Architecture Visualization Tool Survey. Technical Report, sebis, Technical University Munich (2014).
14. Sousa, P. et al.: Enterprise Transformation: The Serasa Experian Case. In: Practice-Driven Research on Enterprise Transformation (PRET 2011). LNBIP, vol 89, pp. 134-145. Springer, Berlin, Heidelberg (2011).
15. Sousa, P., Cardoso, D., Colaço, J.: Managing multi-view business processes models in the Atlas tool. In: Proc. of the 19th Enterprise Engineering Working Conference Forum, vol. 2408. CEUR-WS.org (2019).
16. Sousa, P., Lima, J., Sampaio, A., Pereira, C.: An Approach for Creating and Managing Enterprise Blueprints: A case for IT Blueprints. In: 21st International Conference on Advanced Information Systems. LNBIP, vol. 34, pp. 70-84. Springer-Verlag (2009).
17. Sousa, P., Sampaio, A. Leal, R.: A Case for a Living Enterprise Architecture in a Private Bank. In: 8th Workshop on Transformation & Engineering of Enterprises (TEE 2014), Vol 1182. CEUR-WS.org (2014).
18. Sousa, P., Carvalho, M.: Dynamic Organization's Representation. Linking Project Management with Enterprise Architecture. In: IEEE 20th Conf. on Business Informatics (CBI), vol 2, pp. 170-174. IEEE (2018).
19. Sousa, P.; Leal, R.; Sampaio, A.: Atlas: The Enterprise Cartography Tool. In: 18th Enterprise Engineering Working Conference Forum, Vol. 2229. CEUR-WS.org (2018).
20. Steuer, J.: Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4), 73-93 (1992)
21. Tribolet, J., Sousa, P., Caetano, A.: The Role of Enterprise Governance and Cartography in Enterprise Engineering. *Enterprise Modelling and Information Systems Architectures*, 9 (1): 38-49 (2014).
22. van den Berg, M.: ArchiMate, BPMN and UML: An approach to harmonizing the notations. Orbus Software white paper (2012).