An Authoring Tool for Mixed Reality Interfaces for Digital Twins in Manufacturing

Aubrey Simonson[1][0000−0001−7778−5940] and Guodong Shao[2][0000−0001−8856−3622]

[1] Northeastern University, Boston MA 02115, USA
simonson.au@northeastern.edu
[2] Systems Integration Division, Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
guodong.shao@nist.gov

Abstract. While the concept of digital twins has a history of at least twenty years, the design of user interfaces for digital twins is less well studied. Extended Reality (XR) interfaces show particular potential in this domain. Creating a custom XR interface for a digital twin requires the involvement of software engineers with diverse skill sets. Existing efforts to create XR interfaces for digital twins have been case studies designed for the specific use cases they were tested on. This paper introduces an authoring tool that allows end users without programming expertise to create XR interfaces for digital twins without writing any code themselves. Our initial prototype demonstrates the ability to parse machine data formatted according to the MTConnect standard, and provides menu-based interfaces and a sandbox of data visualizations to guide users through interpreting and displaying data.

Keywords: Digital Twin · Mixed Reality · Authoring Tool.

1 Introduction

Within the context of manufacturing, a digital twin is a "fit for purpose digital representation of an observable manufacturing element with synchronization between the element and its digital representation" [12]. Digital twins can help lower costs, improve worker safety, and positively impact the performance of a manufacturing system with respect to productivity, energy consumption, and product quality.

The existing literature contains an abundance of case studies of digital twins [17,6,8,1,5,21,22]. These case studies are usually one-off prototypes that describe how they were created, but are not generic or reusable in other contexts [10]. Current methods for creating digital twins require considerable investment of both funds and human resources.

This paper will introduce a tool which exists in Mixed Reality (MR). Within this paper, we will use XR as an umbrella term to refer to any combination of virtual reality (VR), augmented reality (AR), and mixed reality (MR). We will use MR to refer to the more specific category of virtual information seen while wearing a head-mounted display (HMD) with video see-through that possesses a high level of local presence [20].

XR is a promising medium for interfaces to digital twins in a manufacturing context [23]. AR supports the creation of world-fixed information, or information superimposed at a specific point in 3D space. World-fixed information reduces the cognitive load of understanding information by avoiding context switching between diagrams and the objects they represent [9]. Many modern XR devices are HMDs, which do not require controllers, and instead use hand-tracking and gestures as input devices. Users therefore do not have to halt maintenance work to use interaction tools such as keyboards, touch screens, or printed documents in order to interact with the system [24].

XR environments can also improve the ability of an end user to interpret data. For example, they allow users to see and understand the 3D shapes of objects that are too large or otherwise obfuscated to be observed in physical reality [18]. For data with high-dimensionality, virtual immersive environments allow more variables to be presented simultaneously, using properties such as XYZ spatial coordinates, length, width, height, shape, color, and transparency of the representation of any data point [7]. Henry et al. [11] found that participants exploring a network graph identified properties such as the number of nodes of the same type and the number of neighbors of a particular node more quickly when viewing an immersive, egocentric visualization than when viewing a non-immersive rotateable 3D model of the graph. Exploring multidimensional datasets in immersive VR increases the depth of insights that users gain without increasing their workload [15]. For complex 3D graphs, the stereoscopic vision and motion parallax created by being in an immersive environment improves the visibility of separate points [4].

XR interfaces also open opportunities for computer-supported cooperative work, such as allowing conversations between a physically present user in AR, and a physically remote user in VR, who can both see one another and the same virtual data. One well-documented use case for this type of remote cooperative work is allowing remote experts to assist physically present operators with complex maintenance tasks [1]. Balakrishnan et al. [3] suggests that a visualization tool enabling both physically present and remote users to have edit access improves performance on complex tasks.

Creating and maintaining a custom VR or AR interface for a digital twin (DT) requires both one or more software engineers capable of creating a DT, and one or more software engineers capable of creating an XR experience. In this paper, we propose a potential solution for creating MR visualizations of and interfaces to DTs without incurring the expense involved in creating and maintaining a custom MR application. Our solution is MiRIAD (Mixed Reality Interface Authoring Tool for Digital Twins), a simple authoring tool for creating MR interfaces from MTConnect data. It is designed for use by non-programmers, and has an intuitive and conversational interface.

The rest of this paper is organized as follows: Section 2 describes existing XR DTs. Section 3 describes the MiRIAD prototype, focusing on several practical details of its implementation. Section 4 illustrates the usage of MiRIAD by assembling an MR interface for a digital twin of a UR5e robot arm with it. Section 5 discusses the limitations of this work, and future work.

2 Related Work

Several case studies exist which represent properties of an individual Observable Manufacturing Element (OME) in an AR environment [12]. For example, Zhu et al. [25] used a Microsoft Hololens to visualize the axis position data, machine tool status, cutting tool data, historical tool path data, cutting force data, safety volume data, spindle speed, feed rate, and cutting force of a computer numerical control (CNC) machine. Cai et al. [6] used AR to display where to place robot arms in reconfigurable additive manufacturing scenarios, as determined by a DT.

Much work at the intersection of digital twins (DTs) and VR prototype use-cases relates to human-robot interaction (HRI). VR-based HRI interfaces allow users to program and operate robots without being physically near them. This addresses a number of safety concerns involved in HRI. It also allows users to understand the position of the robot and other objects relative to one another in 3D space, a feature that is useful for teleoperation. For example, Dobot Magician is a VR interface that allowed users to hand-guide a robot arm from the safety of a virtual environment, rather than doing so while being in the same room as the robot arm and interacting with it directly [22]. Wang et al. [21] used a VR interface to allow students who were not experienced carpenters to program a KUKA mobile industrial robotic arm to install drywall panels in an imperfect setting. In their first session using the system, students and the robot were able to install simulated drywall panels faster than experienced carpenters are able to when working alone. Several other VR applications have been implemented which allow users to program robots in virtual reality [5,2,14,21].

Authoring tools for MR applications in manufacturing settings usually require a software engineer or content developer to create content on an application other than the HMD, rather than allowing end users to provide meaningful input while using the application itself. One notable exception is Authorable Context-aware AR System (ACARS). ACARS allows maintenance technicians to author AR contents on-site [24]. ACARS still requires AR content developers to create content via a desktop interface, but allows technicians to modify this content from within the application. Another is Corsican Twin, a tool for authoring augmented reality data visualisations in VR using digital twins [19]. In a user study, Prouzeau et al. [19] found that "Participants in all groups agreed that allowing different end-users to author and customise their own visualisations was important, since each technician is sensitive to the particular types of information they need when diagnosing a failure."

While a number of XR interfaces for DTs have been developed and evaluated, they have each been made by a team of experts, and for a specific use case. Limited explorations that allow end-users to develop XR DTs themselves suggest that end-user authorship may be a valuable approach. This paper introduces an authoring tool that will allow end-users to develop MR interfaces to DTs. At this point in time, authoring tools for MR interfaces to DTs are under-explored in the literature.

3 Prototype

MiRIAD is an authoring tool that allows end users to create MR interfaces for digital twins. The initial prototype described in this section demonstrates the ability to parse machine data formatted according to the MTConnect standard, and provides a number of menu-based interfaces and a sandbox of data visualizations to guide users through interpreting and analyzing data [16].

The flow of control between a user and the application is depicted in Figure 1. The user is first required to provide the application with correctly formatted data. MiRIAD then parses that data into a generic structure used internally, and presents it to the user as a more human-readable series of menus and summary statistics. From these menus, the user selects aspects of the data to include in the visualization, and appropriate visualization tools to represent them. The user is then able to place them in a persistent location in 3D space. While MiRIAD continues to display these visualizations, it does not, in its current form, make decisions or recommendations based on the data. Decision making continues to be the responsibility of users. A system architecture diagram of MiRIAD is shown in Figure 2.

3.1 Data Parsing

MTConnect was chosen as the first data format for MiRIAD to be able to display because of its ease of use, broad adoption, and open-source nature. MTConnect is a protocol for logging events during the operation of a manufacturing machine. "Agents" translate data from equipment, which is frequently in proprietary and non-standard formats, into data that adheres to the MTConnect Standard.

MTConnect data is in Extensible Markup Language (XML). XML is a markup language. MiRIAD is made in the Unity3D game engine, which is comprised primarily of C#, an object-oriented programming language. MiRIAD therefore parses provided MTConnect data to reorganize it into an object-oriented format which it uses internally. This format is a simple tree-structure of nodes, which retains the tree structure from the XML data.

While the MTConnect standard contains an extensive semantic data model, our parser is programmed to recognize only a limited number of relevant keywords. These keywords are "DeviceStream", "ComponentStream", and "Samples". XML nodes with these keywords are represented as more specific node types, in order to be recognized by the user interface (UI) system. Those node types are Devices, Components, and SampleTypes. All node types inherit from the Abstract Node class, which allows for the addition of more node types. XML nodes that do not contain a specific keyword are assigned AbstractNodes.

Samples are treated as a unique case. Data recorded in just a few minutes from just one OME can contain thousands of samples. We therefore do not create a separate node for each sample. Instead, samples with the same name that are children of the same component are aggregated into "SampleTypes". SampleTypes retain summary statistics about samples, such as the maximum value, minimum value, mean of all values, most recent value, and timestamp of the most recent value.

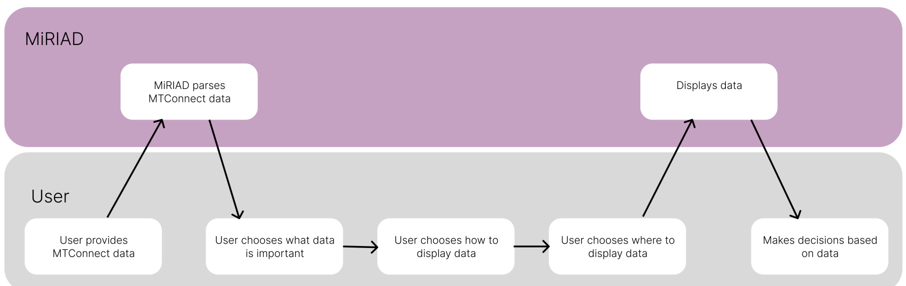


Fig. 1. Flow of control between MiRIAD and an end user. Users provide correctly formatted data to the application, choose what data is important, how to display data, where to display data, and what decisions should be made based on the data. The application parses the data into a more human-readable format, and displays data within a mixed reality environment.

3.2 User Interface

After data has been parsed into the generic structure that MiRIAD uses internally, data is presented to a user as a more human-readable series of menus and summary statistics. This series of menus does not interact with the MTConnect parsing structure in any way– it only relies on data contained within the generic internal structure. This separation is to more easily allow the MTConnect parser to be exchanged for any other parser for use with any other form of hierarchy-based data.

When users enter the virtual environment, they see a menu which lists all devices found in the provided data. Each device has options to either "Edit" or "See More". Edit buttons of Devices open sub-menus of Components, which each have their own "Edit" and "See More" buttons. Edit buttons of Components open sub-menus of Sample Types. Edit buttons of Sample Types display summary statistics of the samples associated with that Sample Type, as well as a list of potential representations with which to visualize them. An example of one summary statistics panel is shown in Figure 3. Any Device, Component, Sample Type, or Representation can be grabbed with the user's hands and moved in 3D

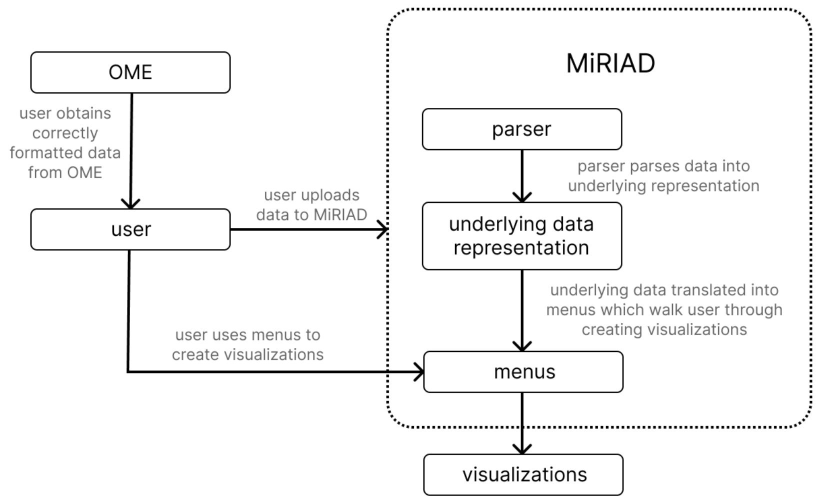


Fig. 2. System architecture diagram of MiRIAD.

space. Removing an item from a menu causes it to be considered "part of the visualization", and therefore controlled by "See More / See Less" buttons, rather than part of the Edit menu. This collapsing tree structure balances reducing visual clutter against specificity of information, by allowing users to temporarily expand sections of the visualization that are relevant, and close parts of the visualization that are in the way.

3.3 Visualizations

This prototype did not include 3D models of the machine itself, as the machine itself is still visibe in MR, and because representations of the physical components of OMEs and their movements are already explored in existing literature [17]. Instead, this work focused on representing generic data types, and displaying the variety of information which would not immediately be visually obvious to a physically-present operator. Visualizations can be grabbed and moved around, and are part of the collapsible menu structure in the same way that other parts of the UI are. Our current menu of visualizations only works for the "float" data type. The following three types of visualization exist in this authoring tool:

Simple Float The simple float representation is one number, presented as text. It is always set to the most recent value.

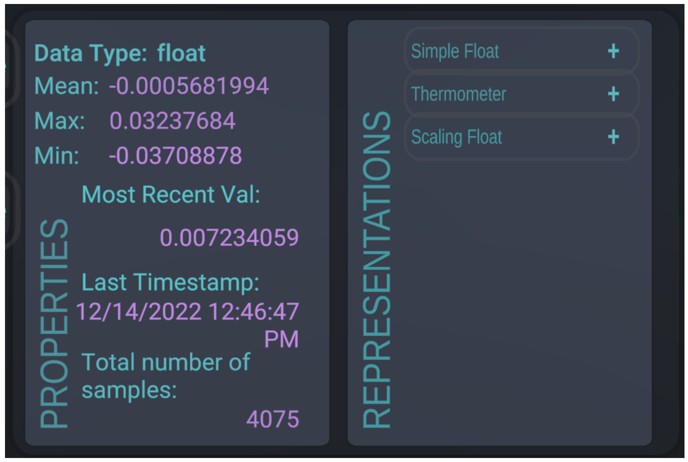


Fig. 3. An example of a summary statistics panel. This panel contains both summary statistics about all samples labelled with the same name that are a child of the same component in the provided data. It also contains a number of options for representations that can be used to represent data summarized on the panel.

Thermometer The thermometer (Figure 4) presents a visual summary of the relationship between all collected summary statistics, other than the timestamp of the most recent sample. The black ring represents the position of the mean relative to the minimum and maximum values. The ratio of the magenta section of the thermometer relative to the total size of the thermometer represents the value of the most recent sample, much like the level of mercury in a physical analog thermometer. Being a round, 3-dimensional object, rather than a flat display, allows this visualization to both be visible from all angles, and cause relatively little occlusion.

Scaling Float The scaling float representation is an exploration of how the unstructured nature of an MR interface can be used to prioritize information based on importance. In the absence of a firmly defined screen with limited space, UI elements have much more freedom to vary in scale while remaining legible. This representation looks much like the simple float, but changes scale based on the most recent value. The scaling float is smaller when the most recent value is closer to the mean of all samples for this sample type, and larger when the most recent sample value is farther from the mean. Assuming that some variables are more important to pay attention to when they are farther from the mean value of the data, this representation adapts the UI to changing conditions with a simple and easy-to-understand mechanic.

The implementation of this research work can be found on its github respository.

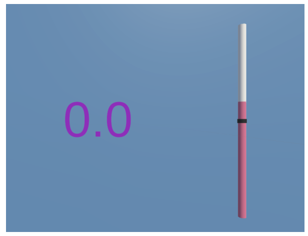


Fig. 4. Simple float and thermometer visualizations. Simple floats only display the value of the most recent sample. Thermometers express how the most recent sample relates to the mean and range of all samples for that sample type. The black ring represents the position of the mean relative to the minimum and maximum values

4 Illustrative Case

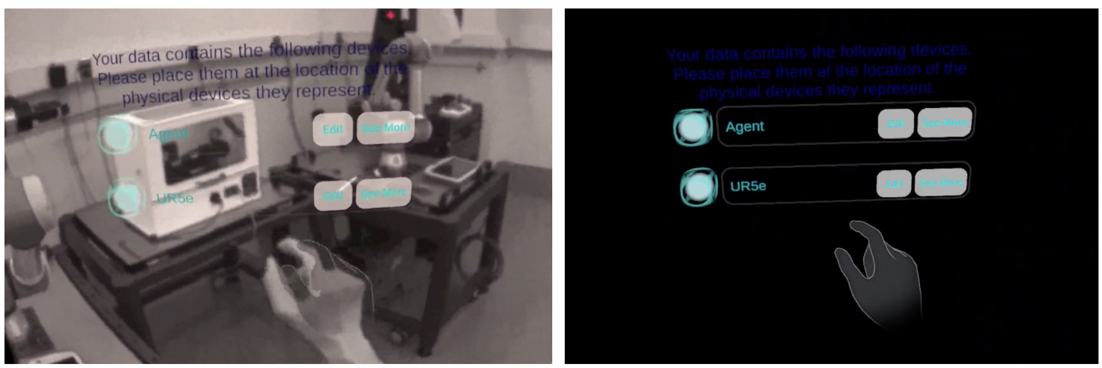
MiRIAD, as an authoring tool, is capable of making many different interfaces from many different sources of data. In this section, to illustrate how MiRIAD might be used, one interface is made from one source of data. In this illustrative case, an XR visualization of data from a UR5e robot arm is created using MiRIAD. This robot arm is part of the Digital Twin Lab (DTL) at the National Institute of Standards (NIST), a workcell consisting of small collaborative robot arms for material handling and machine tending, a computer numerical control (CNC) machine tool for cutting a part, and a coordinate-measuring machine (CMM) for product geometry measurements and quality control[13].

Figures 5-11 are all screencaptures from the process of using MiRIAD. MiRIAD uses passthrough, and runs on a Meta Quest 2. Passthrough is a feature of many recent consumer and commercial-grade HMDs, which allows the user to see a real-time reconstruction of the physical environment while wearing an opaque display. For privacy and security reasons, high-quality screen recordings of passthrough applications only show virtual content– they do not also record the physical environment. Figures 5-11 therefore provide a lower-quality image recorded using scrcpy, a tool for mirroring Android devices, alongside a higher-quality image of the generated content.

Data recorded from the DTL is included with the project before it is built in Unity and uploaded to a Meta Quest 2 HMD. MiRIAD parses this data, and presents the user with a menu of all available devices, as can be seen in Figure 5. From this list of devices, the UR5e is selected.

Many properties of the robot arm are visible in the HMD, such as the angle of any joint, or status of the gripper. This MR visualization will represent the torque of three rotaries, which is more difficult to perceive. Selecting the edit button opens a menu of all components of the robot arm that were found in the data. This components menu is shown in Figure 6. Three rotaries are selected by grabbing them with the hands and removing them from the edit menu. The controller is also selected, so that accumulated time can be used as a debugging tool, to check if new data is being received from the robot arm if none of the torque values are changing.

Each of the three rotary visualizations are placed next to the rotaries they represent, as shown in Figures 7 and 9. On each rotary, pressing "edit" opens the sample types menu. From each of these menus, "torque" is selected by grabbing it and removing it from the edit menu. Selecting the "edit" button on any of these sample types opens the summary statistics panel. Each summary statistics panel includes the maximum value, minimum value, mean value, and most recent value found in the recorded data, as well as the timestamp associated with the most recent value, and the total number of samples found associated with this component and this name. The value of "torque" for each rotary is a float. The summary statistics panels therefore include three possible means of representing a float.

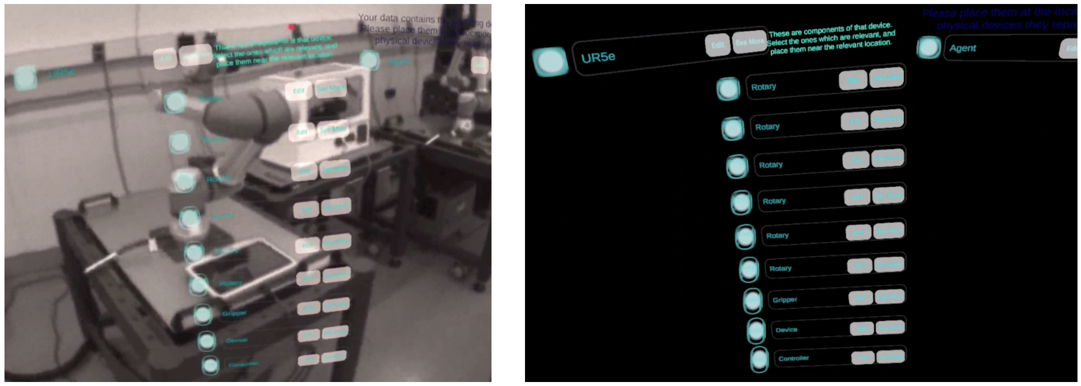


inside of the HMD as recorded via scrcpy Oculus record tool

Fig. 5. When the user first enters the scene, they see a menu of all available devices.

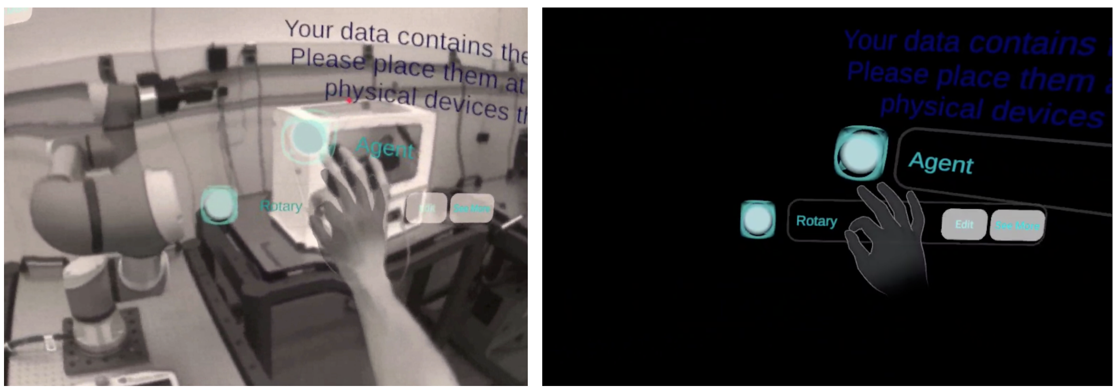
Torque may be more important if the value is unusually high. A scaling float, which will be larger if the value is farther from the mean found in the data, is therefore selected. A scaling float will be large if the value of torque is unusually low as well as unusually high. Therefore, a thermometer, which displays how the most recent value compares to the mean and range of values, is selected in order to help differentiate these cases. This process is then repeated for all three rotaries.

From the controller (a component), accumulated time (a sample type) is selected. Accumulated time is useful as a debugging tool, as it should increase while new data is being received from the robot arm even if none of the torque values are changing. This value is expected to increase monotonically, and the fact that the value is changing is more relevant than what the value is. A simple float is therefore appropriate. Once these data representations have been selected and placed, the visualization (Figure 10 and 11) is finished. It displays information in XR that cannot be seen without it, and does not require any programming on



inside of the HMD as recorded via scrcpy — Oculus record tool

Fig. 6. Selecting the edit button on any devices opens the components menu.



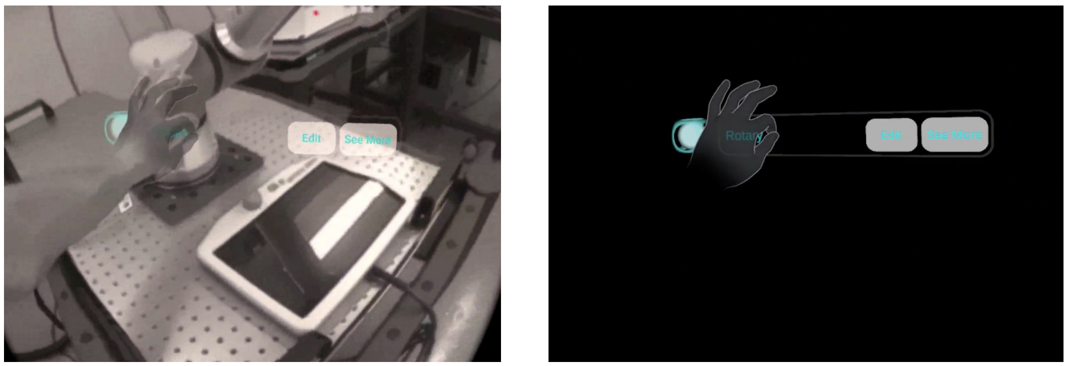
inside of the HMD as recorded via scrcpy — Oculus record tool

Fig. 7. Placing UI for a component next to the physical component it represents.

inside of the HMD as recorded via scrcpy — Oculus record tool

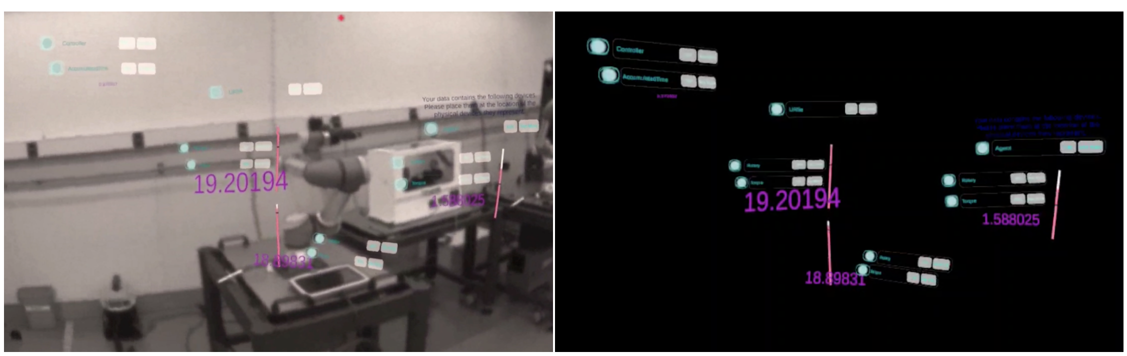
Fig. 8. Creating representations from the representations menu.

the part of an end user to create from the prerecorded data. While, in this case study, we demonstrate making only one interface from only one source of data, MiRIAD, as an authoring tool, is capable of making many different interfaces from many sources of data. Similar interfaces can be made using the same tool from any MTConnect data.



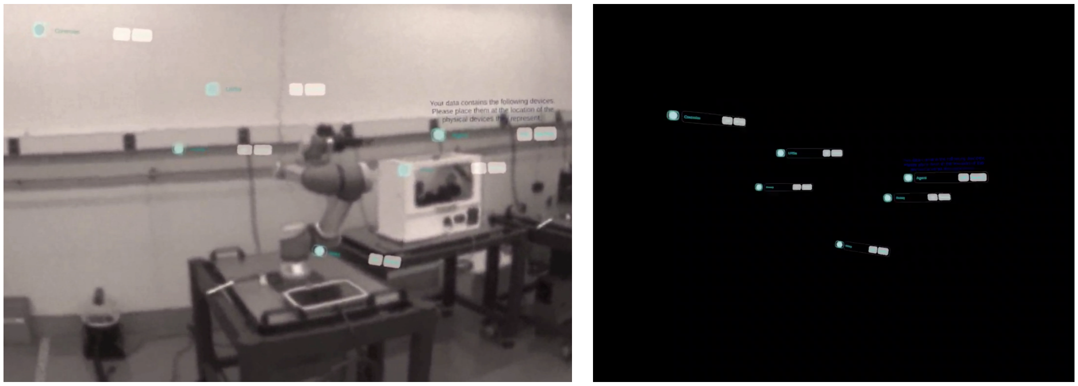
inside of the HMD as recorded via scrcpy Oculus record tool

Fig. 9. Placing a representation near the component it represents. The UI for this component is placed such that it does not occlude the teach pendant, which is also visible.



inside of the HMD as recorded via scrcpy Oculus record tool

Fig. 10. This MR visualization represents the torque of three rotaries, as well as the accumulated time according to the controller.



inside of the HMD as recorded via scrcpy   Oculus record tool

Fig. 11. To reduce occlusion, the visualization for each component can be hidden using the "See Less" button.

5 Discussion

The use of MiRIAD significantly reduces the need for specialized programming expertise, thereby making XR interfaces more accessible. Creating an XR interface, as described in Section 4, could be done by the end-user of the interface, rather than a dedicated software engineer, allowing greater customization to their specific needs. Consequently, MiRIAD represents a significant stride towards enabling manufacturing systems to harness the benefits of XR interfaces and DTs.

In its present state, MiRIAD can be seen as a preliminary exploration rather than a fully functional prototype. MiRIAD has not yet undergone evaluation via user testing, which is important future work. The tool currently lacks certain features, such as the capability to remove items from the interface or to save designed interfaces. Additionally, MiRIAD's current version is limited to representing float values and only accommodates one float per visualization, rather than ways to represent the interaction between variables. MiRIAD can be used to generate an interface from pre-recorded data, but does not have the ability to display real-time data with that interface. These identified limitations present valuable avenues for future research and development. While MiRIAD currently can only represent MTConnect data, its extensible underlying data structure is designed to also incorporate data from a myriad of other sources. Other possible future work includes expanding MiRIAD to work on other HMDs. Figures 5-11 are recorded on a Meta Quest 2, one of the earliest HMDs to use passthrough. More recent HMDs offer higher resolution and passthrough in color.

6 Conclusion

In this paper we present MiRIAD, an authoring tool for creating mixed reality interfaces for digital twins. MiRIAD allows end users without programming expertise to create XR interfaces for digital twins without writing any code themselves. This allows users to experience the benefit of digital twins with XR interfaces. Our initial prototype demonstrates the ability to parse machine data formatted according to the MTConnect standard, and provides a number of menu-based interfaces and a sandbox of data visualizations to guide users through interpreting and displaying data. Use of this prototype is demonstrated by creating an MR interface for a UR5e robot arm. This initial body of work demonstrates the feasibility of authoring tools for MR interfaces to DTs.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Disclaimer

No endorsement of any commercial product by NIST is intended. Commercial materials are identified in this report to facilitate better understanding. Such identification does not imply endorsement by NIST nor does it imply the materials identified are necessarily the best for the purpose.

References

1. Al-Yacoub, A., Eaton, W., Zimmer, M., Buerkle, A., Ariansyah, D., Erkoyuncu, J.A., Lohse, N.: Investigating the impact of human in-the-loop digital twin in an industrial maintenance context (2020)
2. Arnarson, H., Solvang, B., Shu, B.: The application of virtual reality in programming of a manufacturing cell. In: 2021 IEEE/SICE International Symposium on System Integration (SII). pp. 213–218. IEEE (2021)
3. Balakrishnan, A.D., Fussell, S.R., Kiesler, S.: Do visualizations improve synchronous remote collaboration? In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1227–1236 (2008)
4. Brunhart-Lupo, N., Bush, B.W., Gruchalla, K., Smith, S.: Simulation exploration through immersive parallel planes. In: 2016 Workshop on Immersive Analytics (IA). pp. 19–24. IEEE (2016)
5. Burghardt, A., Szybicki, D., Gierlak, P., Kurc, K., Pietruś, P., Cygan, R.: Programming of industrial robots using virtual reality and digital twins. Applied Sciences **10**(2), 486 (2020)
6. Cai, Y., Wang, Y., Burnett, M.: Using augmented reality to build digital twin for reconfigurable additive manufacturing system. Journal of Manufacturing Systems **56**, 598–604 (2020)
7. Donalek, C., Djorgovski, S.G., Cioc, A., Wang, A., Zhang, J., Lawler, E., Yeh, S., Mahabal, A., Graham, M., Drake, A., et al.: Immersive and collaborative data visualization using virtual reality platforms. In: 2014 IEEE International Conference on Big Data (Big Data). pp. 609–614. IEEE (2014)

8. Falah, M.F., Sukaridhoto, S., Al Rasyid, M.U.H., Wicaksono, H.: Design of virtual engineering and digital twin platform as implementation of cyber-physical systems. Procedia Manufacturing **52**, 331–336 (2020)
9. Gattullo, M., Evangelista, A., Uva, A.E., Fiorentino, M., Gabbard, J.L.: What, how, and why are visual assets used in industrial augmented reality? a systematic review and classification in maintenance, assembly, and training (from 1997 to 2019). IEEE transactions on visualization and computer graphics **28**(2), 1443–1456 (2020)
10. Hanke, A., Vernica, T., Bernstein, W.Z.: Linking performance data and geospatial information of manufacturing assets through standard representations. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. vol. 83983, p. V009T09A063. American Society of Mechanical Engineers (2020)
11. Henry, J.A., Polys, N.F.: The effects of immersion and navigation on the acquisition of spatial knowledge of abstract data networks. Procedia Computer Science **1**(1), 1737–1746 (2010)
12. ISO: Automation systems and integration—digital twin framework for manufacturing—part 1: overview and general principles. International Organization for Standardization Geneva, Switzerland (2021)
13. Kibira, D., Shao, G., Venketesh, R.: Building a digital twin of an automated robot workcell. In: 2023 Annual Modeling and Simulation Conference (ANNSIM). pp. 196–207. IEEE (2023)
14. Li, X., He, B., Zhou, Y., Li, G.: Multisource model-driven digital twin system of robotic assembly. IEEE Systems Journal **15**(1), 114–123 (2020)
15. Millais, P., Jones, S.L., Kelly, R.: Exploring data in virtual reality: Comparisons with 2d data visualizations. In: Extended abstracts of the 2018 CHI conference on human factors in computing systems. pp. 1–6 (2018)
16. MTConnect Institute: Mtconnect standard, part 1.0 – overview and fundamentals version 1.8.0. Association for Manufacturing Technology (2021)
17. Pérez, L., Rodríguez-Jiménez, S., Rodríguez, N., Usamentiaga, R., García, D.F.: Digital twin and virtual reality based methodology for multi-robot manufacturing cell commissioning. Applied sciences **10**(10), 3633 (2020)
18. Porcino, T.M., Dórea, M.M., Barboza, D., Oliveira, W., Romani, E., Perin, F., Batista, J.H.: A real-time approach to improve drilling decision-making process using virtual reality visualization. In: 2021 IEEE conference on virtual reality and 3d user interfaces abstracts and workshops (VRW). pp. 755–756. IEEE (2021)
19. Prouzeau, A., Wang, Y., Ens, B., Willett, W., Dwyer, T.: Corsican twin: Authoring in situ augmented reality visualisations in virtual reality. In: Proceedings of the international conference on advanced visual interfaces. pp. 1–9 (2020)
20. Rauschnabel, P.A., Felix, R., Hinsch, C., Shahab, H., Alt, F.: What is xr? towards a framework for augmented and virtual reality. Computers in human behavior **133**, 107289 (2022)
21. Wang, X., Liang, C.J., Menassa, C.C., Kamat, V.R.: Interactive and immersive process-level digital twin for collaborative human–robot construction work. Journal of Computing in Civil Engineering **35**(6), 04021023 (2021)
22. Yigitbas, E., Karakaya, K., Jovanovikj, I., Engels, G.: Enhancing human-in-the-loop adaptive systems through digital twins and vr interfaces. In: 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). pp. 30–40. IEEE (2021)

23. Yin, Y., Zheng, P., Li, C., Wang, L.: A state-of-the-art survey on augmented reality-assisted digital twin for futuristic human-centric industry transformation. Robotics and Computer-Integrated Manufacturing **81**, 102515 (2023)
24. Zhu, J., Ong, S.K., Nee, A.Y.: An authorable context-aware augmented reality system to assist the maintenance technicians. The International Journal of Advanced Manufacturing Technology **66**, 1699–1714 (2013)
25. Zhu, Z., Liu, C., Xu, X.: Visualisation of the digital twin data in manufacturing by using augmented reality. Procedia Cirp **81**, 898–903 (2019)