JUIC-IoT: Just-In-Time User Interfaces for Interacting with IoT Devices in Mixed Reality

Lucien Ledermann
University of St. Gallen
St.Gallen, Switzerland
lucien.ledermann@student.unisg.ch

Jannis Strecker-Bischoff
University of St. Gallen
St.Gallen, Switzerland
jannis.strecker-bischoff@unisg.ch

Kimberly Garcia
University of St. Gallen
St.Gallen, Switzerland
kimberly.garcia@unisg.ch

Simon Mayer
University of St. Gallen
St.Gallen, Switzerland
simon.mayer@unisg.ch

Abstract

The number of deployed Internet of Things (IoT) devices is continuously increasing. While Mixed Reality (MR) allows hands-free interaction, creating MR User Interfaces (UI) for each IoT device is challenging, as often a separate interface has to be designed for each individual device. Additionally, approaches for *automatic* MR UI generation often still require manual developer intervention. To address these issues, we propose the JUIC-IoT system, which automatically assembles Just-in-Time MR UIs for IoT devices based on the machine-understandable format W3C Web of Things Thing Description (TD). JUIC-IoT detects an IoT device with object recognition, uses its TD to prompt an LLM for automatically selecting appropriate UI components, and then assembles a UI for interacting with the device. Our evaluation of JUIC-IoT shows us that the choice of LLM and the TD of a device are more crucial than the formulation of the input prompts for obtaining a usable UI. JUIC-IoT represents a step towards dynamic UI generation, thereby enabling intuitive interactions with IoT devices.

CCS Concepts

• **Human-centered computing → Mixed / augmented reality**; **Ubiquitous computing**.

Keywords

adaptive user interfaces, Internet of Things, Web of Things, automatic generation of user interfaces.

ACM Reference Format:
Lucien Ledermann, Jannis Strecker-Bischoff, Kimberly Garcia, and Simon Mayer. 2025. JUIC-IoT: Just-In-Time User Interfaces for Interacting with IoT Devices in Mixed Reality. In *Companion of the the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '25), October 12–16, 2025, Espoo, Finland*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3714394.3754371

https://doi.org/10.1145/3714394.3754371

1 Introduction

The Internet of Things (IoT) is nowadays widely adopted in several sectors, from smart homes equipped with smart appliances, lighting, and security systems [4, 27, 33] to smart cities capable of managing traffic and monitoring environmental conditions [5, 11, 16, 29]. In agriculture, the IoT supports farming by monitoring soil, water, and crop conditions [3, 8, 21], and in industry, IoT devices are used in tasks such as real-time monitoring, predictive maintenance, and production optimization [2, 7, 20, 24]. However, it is not always easy to interact with IoT devices, since they are vastly heterogeneous, locking end users in vendor specific technologies, including UIs, which leads to common scenarios in which a person requires different applications to interact with each IoT device in a *smart* environment. To tackle the heterogeneity of IoT devices, the Web of Things (WoT) [13] was proposed, and Web standards [40] have been created to provide uniform descriptions of the devices, including their capabilities; allowing for interoperability among devices, and facilitating interacting with them.

Mixed Reality (MR) [31] provides a hands-free way for interacting with IoT devices. However, traditional UI development is time-consuming and labor intensive [23], which is not ideal when navigating from one IoT environment to another. Hence, in this work, we explore using generative artificial intelligence (GenAI) to automate the creation of Just-in-Time (JIT) UIs for interacting with IoT devices using MR. JIT refers to the creation of a UI at the exact moment a user interacts with an object, allowing for a more tailored experience. As functionalities of IoT devices may dynamically change, e.g., based on user roles or environmental changes, a JIT-approach enables highly adaptable and contextually relevant interactions with IoT devices without the need for manual intervention. In this contribution, we explore how general-purpose Large Language Models (LLMs) can accelerate the process of creating UIs for interacting with IoT devices based on standardized descriptions (i.e., W3C WoT Thing Descriptions [39]). Concretely, we propose the JUIC-IoT system (pronounced "juice it!"), which uses an object detection algorithm to categorize objects in the field of view of a user wearing a Head Mounted Display (HMD). Then, an LLM is cued to select (among predefined UI elements) the most suitable UI element for a specific interaction with the IoT device. Next, JUIC-IoT assembles appropriate means to trigger selected actions of the IoT device, thereby enabling people to have control over the devices in their physical environment. JUIC-IoT is capable of creating JIT

UI in MR, paving the way to dynamic generation of UIs according the user context (e.g., role, language) and environmental conditions (e.g., lighting, available devices).

2 Related Work

Several efforts have demonstrated promising synergies between MR and the IoT. From MR applications to provide end-users with an immersive experience to control vendor-specific IoT devices (i.e., Tuya) [1], and applications that help users create control flows for managing IoT devices [28]; to applications that consider several protocols to interact with diverse IoT devices [42]. However, given the large amount of IoT solutions and their heterogeneity, there are several recurrent challenges that MR systems face, such as scalability, flexibility, and seamless integration of new IoT devices [6].

To specifically tackle the heterogeneity of IoT devices (aside from MR), the Web of Things (WoT) was proposed [13], whose objective is to bring any type and any size of devices (known as Things) to the Web, by following the Web architecture principles. The WoT as standardized by the World Wide Web Consortium (W3C WoT), proposes the creation of Thing Descriptions (W3C WoT TD), which are machine-readable and -understandable descriptions of the programming interfaces of WoT-enabled devices[1]. These uniform descriptions act as an interoperability layer that enables the creation of systems that communicate seamlessly with WoT-enabled devices. A device's TD specifies its *interactions affordances*, which can be of one of three types: property, interaction, or event. A *property affordance* corresponds to a state that is produced by the WoT device, an *interaction affordance* triggers an action on the device, and an *event affordance* is usually used to subscribe to a device that periodically produces responses. MR applications that take advantage of TDs have been proposed in the context of digital companions to assist users in ubiquitous computing environments [12, 32, 34]. However, the UIs to interact with WoT-enabled devices are often still created manually.

Automatic Creation of User Interfaces for IoT Devices. Several works have explored the automatic creation of UIs to interact with IoT devices. De la Torre et al. [35], proposed the creation of Web UIs from the metadata of Online Labs (OLs), which are considered a particular case of IoT devices. Similar to TDs, a vocabulary is proposed to uniformly describe OLs, including the expected data types on each interaction. This vocabulary was manually mapped to UI elements (e.g., a button and a label) and their corresponding HTML tag. Using this mapping, a UI can be automatically rendered at runtime. Mayer et al. [22] proposed a taxonomy of typical high-level interaction semantics and a scheme for describing WoT devices. This taxonomy is linked to diverse interactors, that can be used to trigger an action (e.g., a knob UI element to modify the light intensity, a swiping motion, or a shaking movement). Hence, an interface can be automatically created given an annotated WoT device. Salama et al. [26] develop a C# library to parse W3C WoT TD, to dynamically create MR UIs based on a specific TD. However, these UIs do not discriminate the type of interaction or the expected data type in a UI element. While these methods facilitate the creation of UIs by annotating IoT and WoT devices with a specific vocabulary, in some cases, they require that developers provide the general metadata of the device, and the UI-relevant metadata.

[1]We refer to WoT-enabled IoT devices as *WoT devices*.

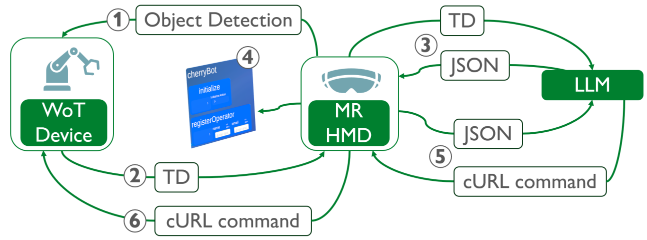


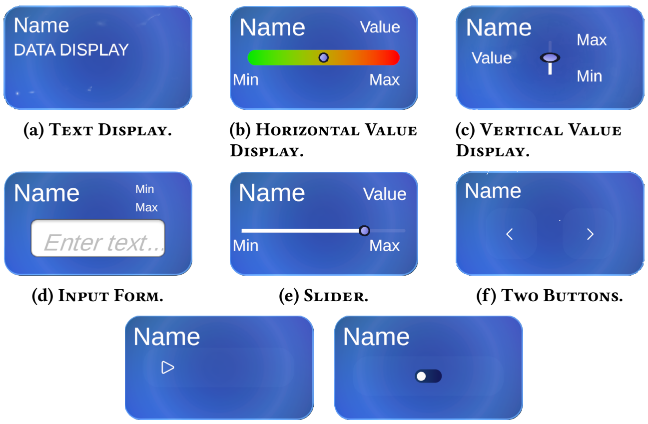
Figure 1: An overview of the JUIC-IoT system comprising one or multiple WoT devices, an MR HMD, and an LLM.

Generative AI (GenAI) for User Interfaces. rGiven the advances and popularization of GenAI, an increasing number of works explore its use for UI design prototyping [25, 30], UI guiding design exploration processes [17], and generating feedback about designs [10]. Hence, the use of Gen-AI tools is mostly focused on the design of UIs rather than on the actual creation of the UI that is then presented to users. GenAI is also increasingly used in MR applications [14], for instance, to augment objects in MR by providing contextual information [9], to allow users to dynamically customize MR environments [36], and to adapt MR UIs considering social and environmental situations [18]. These works focus on the virtual content displayed for a user, or they rely on prior knowledge of the objects users interact with.

Just-in-Time User Interfaces (JIT). GenAI may also be beneficial for the creation of JIT UIs. JIT UI have shown to reduce user errors and cognitive load by delivering just enough information at the right time [15], or to adjust in real-time from learned user behavior [19].

3 Concept and System

Drawing on research from these different fields, we present the JUIC-IoT system as a proof-of-concept. Our system enables people to interact in MR with WoT devices using just-in-time generated UIs. JUIC-IoT consists of an MR application (running on a HMD), WoT devices with their respective TDs, and an LLM. On a conceptual level, JUIC-IoT functions as follows (see Figure 1): When a user looks at a WoT device, an object detection algorithm identifies it and maps it to its URI to load its TD (Step 1). The TD is then divided into its affordances for better LLM processing (Step 2). The MR HMD sends each affordance to the LLM with a prompt to select suitable UI components from a list of defined ones. This prompt is accompanied by a history with examples of previously selected components, to obtain consistent responses (Step 3). The prompt specifies the expected output format (see Listing 2). After receiving LLM responses for each affordance, the MR application assembles the UI components and displays the interface to the user (Step 4; see Figure 3 in Appendix A). A UI component that allows interacting with the WoT device includes a send button. When pressing this button, JUIC-IoT collects all the input values and sends them back to the LLM (along with the affordances) to create a cURL command (Step 5). This command executes an HTTP request that then triggers



(a) Text Display. (b) Horizontal Value Display. (c) Vertical Value Display. (d) Input Form. (e) Slider. (f) Two Buttons. (g) Button. (h) Toggle.

Figure 2: The UI building blocks for Sensor data UI components (a-c), and stateless and stateful UI components (d-h).

an interaction with the WoT device (e.g., moving a robotic arm) (Step 6). As before, a prompt and brief chat history are included to define the task and ensure more reliable results. This second LLM prompt is necessary to generate a well-formed cURL command, as the TD does not precisely specify how the request body should be formatted (e.g., when multiple parameters need to be set).

3.1 UI Building Blocks

For the JIT UI, we defined several UI building blocks in Unity as Prefabs (see Figure 2). Based on the output of the LLM, they are then assembled and displayed to users for a given WoT device (see Figure 3 in Appendix A). Using predefined blocks instead of creating the full UI allows to keep the prototype usable, while the system can later be extended to a finer level. Our UI components are based on Mayer et al.'s model-based interface description for smart objects [22], which categorizes interactions into three types: *sensor data*, *stateless actuators*, and *stateful actuators*. In our system, sensor data (e.g., from a temperature sensor) is displayed using Text Display (Fig. 2a) or Horizontal/Vertical Value Displays (Fig. 2b / 2c). Triggerable stateless actuators whose state cannot queried (e.g., a digital doorbell) are represented by the Button (Fig. 2g). The Two Buttons component (Fig. 2f) is used for "go to" abstractions (e.g., a CD-player's forward-button). Triggerable stateful actuators whose state can be queried (e.g., a dimmable lamp) are handled using multiple abstractions: *set* (Input Form), *set value*/*level*/*set intensity* (Slider, Two Buttons or Input Form), *switch* (Toggle), *position* (Slider), and *move* (Two Buttons).

3.2 Prototype

In our implementation, the main application is running on a Microsoft HoloLens 2, and was created with Unity 2022.3.34 and the Microsoft's Mixed Reality Toolkit 3 (MRTK3). We used a custom trained YOLOv7 [41] model for the object detection. We furthermore implemented the connection to three local LLMs (OpenAI GPT-4o, Google Gemma-2-27, Microsoft Phi-4) and one remotely connected one (OpenAI GPT-4o). The communication with the locally hosted LLMs was managed through LMStudio[2] running on a Windows 11 PC with a a Nvidia GeForce 4070 Ti graphics card. As LM studio provides a local server that mimics the API endpoint of OpenAI, all models could be addressed with the same implementation. We set the temperature to 0.2 for the interaction with the LLMs to create more predictable outputs. Additionally, we used the TDs from four WoT devices in our lab (see Figure 4 in Appendix B): the robotic arm "Cherrybot"[3] (nine affordances), the mobile robot "Tractorbot"[4] (three affordances), smart blinds (two affordances) and a smart light (two affordances). The TDs were in either RDF Turtle format [37] or in JSON-LD [38], as these are both common formats for TDs.[5]

Listing 1: The affordance for operating the Cherrybot's gripper, as it is describe in the robotic arm's TD (in RDF).

```
[ a td:PropertyAffordance , cherrybot:Gripper , js:
IntegerSchema , cherrybot:GripperValue;
      td:name "gripper";
      td:hasForm [
          hctl:hasTarget <https://api.our.labs.website.
com/cherrybot/gripper>;
          hctl:forContentType "application/json";
          hctl:hasOperationType td:readProperty ];
      td:isObservable false;
      js:minimum "0"^^xsd:int;
      js:maximum "800"^^xsd:int ];
```

Listing 2: An optimal JSON response from an LLM for the Cherrybot's gripper's affordance in Listing 1

```
{ "gripper": [ { "numeration": "1",
            "name": "Gripper",
            "uiComponent": "Value Horizontal Display",
            "valueMAX": "800",
            "valueMIN": "0"} ]}
```

4 Evaluation

To evaluate our approach, we focused on the UI selection through the LLM (i.e. Step 3 in Figure 1), as this is the most crucial part of the JUIC-IoT system. Listing 1 provides an example of a property affordance from the Cherrybot's TD, and Listing 2 shows the corresponding expected response from the LLM. Here, the suggested UI component would be a Value Horizontal Display (see Figure 2b). We tested this step of our system's workflow with the setup described in Section 3.2.

4.1 Method

For our evaluation, we tested whether different LLMs and prompts yield significantly different results. We created four prompts to assess how their formulation affects UI building block selection, one manually (PH1), and three prompts were generated with ChatGPT (PAI1, PAI2, and PAI3)[6]. We used ChatGPT to create textual descriptions for all UI building blocks (see Figure 2)) and to develop the remaining prompts, supplying the desired input (see Listing 1) and output formats (see Listing 2).

[2]https://lmstudio.ai/. Last accessed May 15, 2025.
[3]Ufactory xARM 7, see: https://www.ufactory.cc/xarm-collaborative-robot/. Last accessed May 15, 2025.
[4]M5Stack LidarBot, see: https://docs.m5stack.com/en/app/lidarbot. Last accessed May 15, 2025.
[5]See https://github.com/Interactions-HSG/JUIC-IoT for the full TDs for each device.
[6]See https://github.com/Interactions-HSG/JUIC-IoT for the full prompts.

We evaluated system performance using four metrics: Accuracy, Reliability, Completeness, and Response Time. Accuracy assesses UI element selection compared to human judgment (1 point for exact match, 0.5 points for a usable but non-intuitive option). Reliability evaluates JSON response validity (1 point for no repair needed, 0.5 points for automated repair possible via a script, 0 points otherwise)[7]. Completeness scores LLM output based on metadata inclusion and relevance (see Listing 2); penalties are applied for missing components). Response Time measures how quickly the system generates a response in seconds. For each of the WoT devices' TDs, we sent each affordance ten times to each LLM per prompt, to ensure more robust results. All three measurements were then calculated individually for each response.

4.2 Results

Accuracy. Overall, the mean accuracy was 0.54 ($SD = 0.43$). Per Model, GPT-4o was first ($M = 0.64, SD = 0.42$), followed by Phi-4 (see Table C1). A Kruskal-Wallis test showed statistically significant differences between the model's accuracy scores ($H = 105.11, df = 3, p < 0.001$). Pair-wise Wilcoxon rank-sum test showed significant differences for all pairings except for Gemma-2 paired with Phi-4. All prompts scored a similar accuracy between 0.53 and 0.55 (see Table C2), and were thus not significantly different. The accuracy per WoT device was highest for the Tractorbot ($M = 0.77, SD = 0.30$), and lowest for the Blinds ($M = 0.11, SD = 0.20$; see Table C3).

Reliability. The overall mean Reliability score was 0.51 ($SD = 0.07$; scale from 0 to 1). Across all categories, it was on average between 0.50 and 0.56, as the vast majority had a Reliability score of 0.5 (see Table C2). This means most often the LLMs do not return the response in a fully valid JSON-format but with minor, easy to correct syntax mistakes. Only the response to PH1 got a score of '1' in 72 iterations with either GPT-4o or Llama-3.

Completeness. The overall Completeness score was on average 0.65 ($SD = 0.36$; scale from 0 to 1). It was highest for the GPT-4 responses ($M = 0.8, SD = 0.29$), followed by Phi-4 (see Table C1), with significant differences between them ($H = 342.84, df = 3, p < 0.001$). Pair-wise Wilcoxon rank-sum test showed significant differences again for all pairings except for Gemma-2 paired with Phi-4. Per prompt, the AI generated prompt scored between 0.70 and 0.73, while the human-generated one, PH1, scored only 0.66 on average ($SD = 0.30$; see Table C2), with significant differences between the prompts ($H = 88.84, df = 3, p < 0.001$). Pair-wise Wilcoxon rank-sum tests showed significant differences only for the pairings of PH1 with each of the other ones respectively.

Response Time. On average the Response Time was 15.76s ($SD = 21.26$). Gemma-2's response time was surprisingly high ($M = 48.69$, $SD = 26.39$), while the setup with the other LLMs responded below 10s on average (see Table C1). Disregarding Gemma-2, the mean Response Time for PH1 (the shortest prompt) was only 2.32s ($SD = 1.80$), and around four times as long for the AI prompts. Per WoT device, the Response Times were significantly different ($H = 41.89, df = 3, p < 0.001$; no Gemma-2), but all between 5.79s (Lights; two affordances) and 7.51s (Roboticarm; nine affordances).

[7]After we completed this project, OpenAI introduced Structured Outputs in their API which always guarantees a reliable output, see https://openai.com/index/introducing-structured-outputs-in-the-api/, last accessed July 15, 2025.

5 Discussion

In general, our evaluation showed that LLMs can produce usable results for matching TD affordances with UI component descriptions. The choice of LLM impacts the *Accuracy* and *Completeness* of responses, with GPT-4o yielding the best average results in our setup. AI-generated prompts outperformed human-generated ones in *Completeness*, suggesting that using an LLM to create prompts enhances the selection of appropriate UI components. Additionally, *Accuracy* varied significantly per WoT device, indicating that LLMs struggle with interpreting some device affordances. This emphasizes the need for well-formulated, unambiguous TDs. Furthermore, we found that the *Response Time* depends on the used LLM, the length of the prompt and number of affordances in a TD. Compared to previous proposals for automatic interface generation for WoT devices such as HoloWoT [26], our systems offers a greater flexibility, as it can dynamically react to changes in a TD on the fly. This promises more intuitive interaction possibilities for people, and provides a space for personalized adaption of the interface. Our system furthermore facilities the interaction with new, unknown WoT devices through its inclusion of TDs, and thus mitigates the need for creating a distinct interface for each individual WoT device.

Our primary aim was to provide a basic proof-of-concept, hence the visual design and layout can be improved. Currently, UI elements are listed one below the other, which may lead to a confusing UI for WoT devices with many affordances. We thus plan to implement more sophisticated mechanisms for arranging UI components. Additionally, our approach is limited to WoT devices, making its applicability dependent on widespread adoption. Furthermore, we plan to enhance our prototype's robustness and conduct a user study to assess users' perceptions of the system.

Finally, JUIC-IoT provides a starting point towards including users' context in the creation of JIT UIs. This would allow, for example, to show only functionalities that are currently relevant for a user (e.g., a robot technician in a floor-shop will only see UI components that help them perform maintenance tasks. While, a supervisor will be able to see components that provide them performance information), or to adapt in real-time the visual design of the UI to the user preferences (e.g., a colorful or a neutral design).

6 Conclusion

In this work, we explored automatically creating just-in-time UIs for interacting with WoT devices through MR. Our JUIC-IoT prototype uses TDs to access device interaction possibilities and communicates with LLMs to match them with pre-defined UI components. The system showcases that general-purpose LLMs, not specifically trained for this task, can select UI components based on TD information. The evaluation showed that the choice of LLM, the quality of TDs, and the input length are more crucial for usable and fast results than prompt formulation. Researchers should thus carefully choose their LLM, ensure TDs have well-formulated affordances, and provide concise LLM inputs.

JUIC-IoT provides a step towards the just-in-time creation of UIs for interacting with WoT devices in MR, enabling people to easily see an interface for a WoT device right when they want to interact with it, without the need for prior knowledge about the device except for its TD.

References

[1] Zacky Maulana Achmad, Sritrusta Sukaridhoto, Maretha Ruswiansari, Cahyo Arissabarno, Achmad Fatrian Romadhoni, Agus Prayudi, and Rizqi Putri Nourma Budiarti. 2023. Integration Mixed Reality with Internet of Things by Using Magic Leap. In *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE, Palembang, Indonesia, 661–666. https://doi.org/10.1109/EECSI59885.2023.10295629

[2] Filipe Alves, Hasmik Badikyan, H.J. António Moreira, João Azevedo, Pedro Miguel Moreira, Luís Romero, and Paulo Leitão. 2020. Deployment of a Smart and Predictive Maintenance System in an Industrial Case Study. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE, Delft, Netherlands, 493–498. https://doi.org/10.1109/ISIE45063.2020.9152441

[3] Richard Charles Andrew, Reza Malekian, and Dijana Capeska Bogatinoska. 2018. IoT solutions for precision agriculture. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, Opatija, Croatia, 0345–0349. https://doi.org/10.23919/MIPRO.2018.8400066

[4] Nirnay Bansal. 2020. IoT Applications in Smart Homes. In *Designing Internet of Things Solutions with Microsoft Azure : A Survey of Secure and Smart Industrial Applications*. Apress, Berkeley, CA, 135–156. https://doi.org/10.1007/978-1-4842-6041-8_8

[5] Simon Elias Bibri. 2018. The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. *Sustainable Cities and Society* 38 (2018), 230–253. https://doi.org/10.1016/j.scs.2017.12.034

[6] Oscar Blanco-Novoa, Paula Fraga-Lamas, Miguel A. Vilar-Montesinos, and Tiago M. Fernández-Caramés. 2020. Creating the Internet of Augmented Things: An Open-Source Framework to Make IoT Devices and Augmented and Mixed Reality Systems Talk to Each Other. *Sensors* 20, 11 (2020), 27. https://doi.org/10.3390/s20113328

[7] Federico Civerchia, Stefano Bocchino, Claudio Salvadori, Enrico Rossi, Luca Maggiani, and Matteo Petracca. 2017. Industrial Internet of Things monitoring solution for advanced predictive maintenance applications. *Journal of Industrial Information Integration* 7 (2017), 4–12. https://doi.org/10.1016/j.jii.2017.02.003

[8] Anil Kumar D B, Doddabasappa N, Bansilal Bairwa, Anil Kumar C S, Ganesh Raju, and Madhu. 2023. IoT-based Water Harvesting, Moisture Monitoring, and Crop Monitoring System for Precision Agriculture. In *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, Ballar, India, 1–6. https://doi.org/10.1109/ICDCECE57866.2023.10150893

[9] Mustafa Doga Dogan, Eric J Gonzalez, Karan Ahuja, Ruofei Du, Andrea Colaço, Johnny Lee, Mar Gonzalez-Franco, and David Kim. 2024. Augmented Object Intelligence with XR-Objects. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 19, 15 pages. https://doi.org/10.1145/3654777.3676379

[10] Peitong Duan, Jeremy Warner, and Bjoern Hartmann. 2023. Towards Generating UI Design Feedback with LLMs. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 70, 3 pages. https://doi.org/10.1145/3586182.3615810

[11] Mauro Fadda, Matteo Anedda, Roberto Girau, Giovanni Pau, and Daniele D. Giusto. 2023. A Social Internet of Things Smart City Solution for Traffic and Pollution Monitoring in Cagliari. *IEEE Internet of Things Journal* 10, 3 (2023), 2373–2390. https://doi.org/10.1109/JIOT.2022.3211093

[12] Kimberly Garcia, Jonathan Vontobel, and Simon Mayer. 2024. A Digital Companion Architecture for Ambient Intelligence. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 2, Article 66 (May 2024), 26 pages. https://doi.org/10.1145/3659610

[13] Dominique Guinard, Vlad Trifa, and Erik Wilde. 2010. A resource oriented architecture for the Web of Things. In *2010 Internet of Things (IOT)*. IEEE, Tokyo, Japan, 1–8. https://doi.org/10.1109/IOT.2010.5678452

[14] Teresa Hirzle, Florian Müller, Fiona Draxler, Martin Schmitz, Pascal Knierim, and Kasper Hornbæk. 2023. When XR and AI Meet - A Scoping Review on Extended Reality and Artificial Intelligence. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–45. https://doi.org/10.1145/3544548.3581072

[15] Stephen S. Intille. 2004. Ubiquitous Computing Technology for Just-in-Time Motivation of Behavior Change. In *MEDINFO 2004*. IOS Press, Amsterdam, NL, 1434–1437. https://doi.org/10.3233/978-1-60750-949-3-1434

[16] Aqeel Kazmi, Elias Tragos, and Martin Serrano. 2018. Underpinning IoT for Road Traffic Noise Management in Smart Cities. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Athens, Greece, 765–769. https://doi.org/10.1109/PERCOMW.2018.8480142

[17] Seung Won Lee, Tae Hee Jo, Semin Jin, Jiin Choi, Kyungwon Yun, Sergio Bromberg, Seonghoon Ban, and Kyung Hoon Hyun. 2024. The Impact of Sketch-guided vs. Prompt-guided 3D Generative AIs on the Design Exploration Process. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 1057, 18 pages. https://doi.org/10.1145/3613904.3642218

[18] Zhipeng Li, Christoph Gebhardt, Yves Inglin, Nicolas Steck, Paul Streli, and Christian Holz. 2024. SituationAdapt: Contextual UI Optimization in Mixed Reality with Situation Awareness via LLM Reasoning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 43, 13 pages. https://doi.org/10.1145/3654777.3676470

[19] Jiming Liu, Chi Kuen Wong, and Ka Keung Hui. 2003. An adaptive user interface based on personalized learning. *IEEE Intelligent Systems* 18, 2 (2003), 52–57. https://doi.org/10.1109/MIS.2003.1193657

[20] Yuehua Liu, Wenjin Yu, Tharam Dillon, Wenny Rahayu, and Ming Li. 2022. Empowering IoT Predictive Maintenance Solutions With AI: A Distributed System for Manufacturing Plant-Wide Monitoring. *IEEE Transactions on Industrial Informatics* 18, 2 (2022), 1345–1354. https://doi.org/10.1109/TII.2021.3091774

[21] Rodrigo Filev Maia, Ibrahim Netto, and Anh Lan Ho Tran. 2017. Precision agriculture using remote monitoring systems in Brazil. In *2017 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, San Jose, CA, USA, 1–6. https://doi.org/10.1109/GHTC.2017.8239290

[22] Simon Mayer, Andreas Tschofen, Anind K. Dey, and Friedemann Mattern. 2014. User interfaces for smart things – A generative approach with semantic interaction descriptions. *ACM Trans. Comput.-Hum. Interact.* 21, 2 (Feb. 2014), 25. https://doi.org/10.1145/2584670

[23] Brad A. Myers and Mary Beth Rosson. 1992. Survey on user interface programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. Association for Computing Machinery, New York, NY, USA, 195–202. https://doi.org/10.1145/142750.142789

[24] Parpala, Radu Constantin and Iacob, Robert. 2017. Application of IoT concept on predictive maintenance of industrial equipment. *MATEC Web Conf.* 121 (2017), 02008. https://doi.org/10.1051/matecconf/201712102008

[25] Savvas Petridis, Michael Terry, and Carrie Jun Cai. 2023. PromptInfuser: Bringing User Interface Mock-ups to Life with Large Language Models. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 237, 6 pages. https://doi.org/10.1145/3544549.3585628

[26] Fady Salama, Idil Sezgin, Ege Korkan, Sebastian Käbisch, and Sebastian Steinhorst. 2024. HoloWoT: A First Step Towards Mixed Reality Digital Twins for the Industrial Internet of Things. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24)*. Association for Computing Machinery, New York, NY, USA, 318–321. https://doi.org/10.1145/3652620.3688243

[27] Laila Salman, Safa Salman, Saeed Jahangirian, Mehdi Abraham, Fred German, Charlotte Blair, and Peter Krenz. 2016. Energy efficient IoT-based smart home. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, Reston, VA, USA, 526–529. https://doi.org/10.1109/WF-IoT.2016.7845449

[28] Ronny Seiger, Romina Kühn, Mandy Korzetz, and Uwe Aßmann. 2021. HoloFlows: modelling of processes for the Internet of Things in mixed reality. *Software and Systems Modeling* 20, 5 (Oct. 2021), 1465–1489. https://doi.org/10.1007/s10270-020-00859-6

[29] Jalpa Shah and Biswajit Mishra. 2016. IoT enabled environmental monitoring system for smart cities. In *2016 International Conference on Internet of Things and Applications (IOTA)*. IEEE, Pune, India, 383–388. https://doi.org/10.1109/IOTA.2016.7562757

[30] Jingyu Shi, Rahul Jain, Hyungjun Doh, Ryo Suzuki, and Karthik Ramani. 2023. An HCI-Centric Survey and Taxonomy of Human-Generative-AI Interactions. https://arxiv.org/abs/2310.07127v2.

[31] Richard Skarbez, Missie Smith, and Mary C. Whitton. 2021. Revisiting Milgram and Kishino's Reality-Virtuality Continuum. *Frontiers in Virtual Reality* 2 (March 2021), 647997. https://doi.org/10.3389/frvir.2021.647997

[32] Janick Spirig, Kimberly Garcia, and Simon Mayer. 2022. An Expert Digital Companion for Working Environments. In *Proceedings of the 11th International Conference on the Internet of Things* (St.Gallen, Switzerland) *(IoT '21)*. Association for Computing Machinery, New York, NY, USA, 25–32. https://doi.org/10.1145/3494322.3494326

[33] Cristina Stolojescu-Crisan, Calin Crisan, and Bogdan-Petru Butunoi. 2021. An IoT-Based Smart Home Automation System. *Sensors* 21, 11 (2021), 23. https://doi.org/10.3390/s21113784

[34] Jannis Strecker, Kimberly García, Kenan Bektaş, Simon Mayer, and Ganesh Ramanathan. 2023. SOCRAR: Semantic OCR through Augmented Reality. In *Proceedings of the 12th International Conference on the Internet of Things* (Delft, Netherlands) *(IoT '22)*. Association for Computing Machinery, New York, NY, USA, 25–32. https://doi.org/10.1145/3567445.3567453

[35] Luis de la Torre, Jesus Chacon, Dictino Chaos, Ruben Heradio, and Rajarathnam Chandramouli. 2023. Using IoT-Type Metadata and Smart Web Design to Create User Interfaces Automatically. *IEEE Transactions on Industrial Informatics* 19, 3 (2023), 3109–3118. https://doi.org/10.1109/TII.2022.3186638

[36] Giacomo Vallasciani, Lorenzo Stacchio, Pasquale Cascarano, and Gustavo Marfia. 2024. CreAIXR: Fostering Creativity with Generative AI in XR environments.

In *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*. 1–8. https://doi.org/10.1109/MetaCom62920.2024.00034
[37] W3C. 2014. RDF 1.1 Turtle - Terse RDF Triple Language. https://www.w3.org/TR/turtle/.
[38] W3C. 2020. JSON-LD 1.1 - A JSON-based Serialization for Linked Data. https://www.w3.org/TR/json-ld/.
[39] W3C. 2023. Web of Things (WoT) Thing Description 1.1. https://www.w3.org/TR/wot-thing-description11/.
[40] W3C. 2025. Web of Things (WoT). https://www.w3.org/WoT/.
[41] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2023. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Vancouver, BC, Canada, 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721
[42] Daniel Wright, Russell Skaggs-Schellenberg, and Shahab Tayeb. 2020. Networked Mixed Reality Framework for the Internet of Things. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE, Vancouver, BC, Canada, 1–7. https://doi.org/10.1109/IEMTRONICS51293.2020.9216397

A MR Screenshot

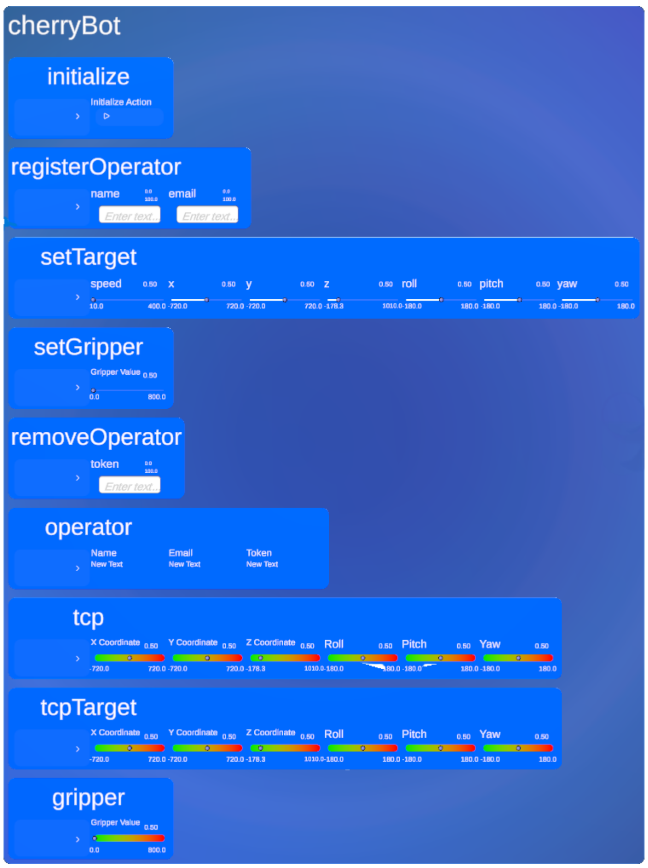
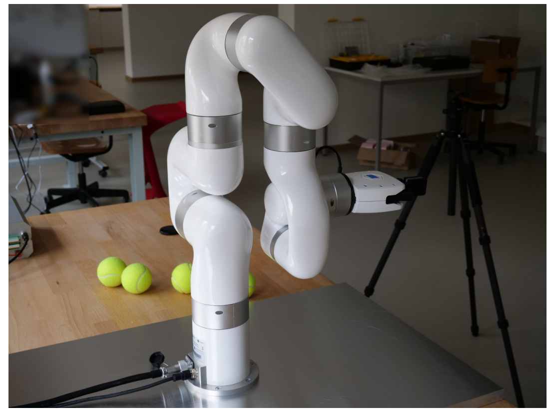
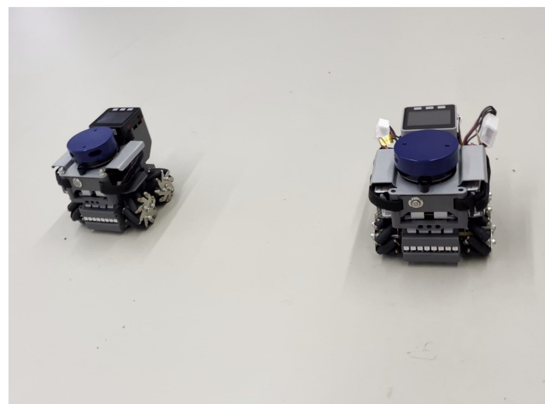


Figure 3: A screenshot of the full UI with UI components for all affordances of the Cherrybot's TD. The user sees this UI in MR on the HL2. The 'Gripper' on the bottom corresponds to the affordance in Listings 1 and 2.

B WoT devices



(a) Robotic arm "Cherrybot"



(b) Mobile robot "Tractorbot"

(c) The Office Lights (front) and Blinds (back)

Figure 4: The WoT devices whose TDs we used for the evaluation.

C Detailed Results of the Evaluation

Table C1: Evaluation results per Model.

Model	Accuracy mean (std)	Reliability mean (std)	Completeness mean (std)	Response Time (s) mean (std)
GPT-4o	0.64 (0.42)	0.51 (0.06)	0.80 (0.29)	7.44 (5.4)
Gemma-2	0.56 (0.46)	0.50 (0)	0.74 (0.36)	48.69 (26.39)
Llama-3	0.40 (0.38)	0.55 (0.15)	0.51 (0.36)	4.93 (2.7)
Phi-4	0.57 (0.41)	0.50 (0)	0.76 (0.31)	8.14 (3.92)

Table C2: Evaluation results per Prompt.

Prompt	Accuracy mean (std)	Reliability mean (std)	Completeness mean (std)	Response Time (s) mean (std)	Response Time (s; w/o Gemma-2) mean (std)
PAI1	0.53 (0.43)	0.50 (0)	0.70 (0.37)	19.19 (23.29)	8.17 (5.19)
PAI2	0.54 (0.42)	0.50 (0)	0.72 (0.37)	20.28 (23.91)	8.47 (3.15)
PAI3	0.55 (0.42)	0.50 (0.00)	0.73 (0.36)	19.91 (23.17)	8.36 (2.89)
PH1	0.54 (0.44)	0.56 (0.16)	0.66 (0.30)	8.95 (17.18)	2.32 (1.80)

Table C3: Evaluation results per WoT Device.

WoT Device	Accuracy mean (std)	Reliability mean (std)	Completeness mean (std)	Response Time (s) mean (std)	Response Time (s; w/o Gemma-2) mean (std)
Blinds	0.11 (0.2)	0.53 (0.12)	0.84 (0.32)	12.84 (12.89)	6.51 (3.39)
Lights	0.42 (0.48)	0.51 (0.08)	0.82 (0.35)	11.62 (12.7)	5.79 (2.94)
Roboticarm	0.59 (0.41)	0.51 (0.07)	0.71 (0.34)	19.57 (25.91)	7.51 (5.05)
Tractorbot	0.77 (0.30)	0.52 (0.09)	0.53 (0.33)	16.67 (21.31)	6.04 (3.2)