NexOz - A Wizard of Oz Approach to Facilitate the Integration of AI in Interactive Systems

Raf Menten ✉[1] [0000-0003-2875-9812], Gustavo Rovelo Ruiz[1] [0000-0001-7580-8950], and Davy Vanacken[1] [0000-0001-8436-5119]

[1]Hasselt University - Flanders Make
Expertise Centre for Digital Media, Diepenbeek, Belgium
{first name}.{last names(s)}@uhasselt.be

Abstract. This paper introduces NexOz, an approach to facilitate incremental integration of AI components into interactive systems using a Wizard of Oz (WOz). The paper explores various challenges of AI integration, such as contextual understanding, conversational interaction, state tracking, and gesture recognition, particularly within the context of the Flanders Make OperatorAssist project in the manufacturing industry. NexOz allows developers to simulate AI functionalities through human intervention, enabling rapid prototyping, experimentation, and data collection. By leveraging human operators to emulate AI behavior, NexOz facilitates the design and testing of dialogue flows, interaction patterns, tracking mechanisms, and conversational designs in real-time, without having a fully-fledged implementation of AI-based solutions. Through iterative refinement and continuous feedback loops, NexOz offers a pragmatic approach to navigate the complexities of AI integration in interactive systems.

Keywords: Wizard of Oz · AI Integration · Engineering Interactive AI-based Systems · Digital Assistant · Industry 4.0 · OperatorAssist.

1 Introduction

As artificial intelligence (AI) continues to advance, there is a growing demand for seamless integration of AI within interactive systems to enhance user experiences, improve system performance, and unlock new capabilities. This integration, however, presents a multifaceted challenge that encompasses both technical and user-centric hurdles [1, 9], ranging from algorithmic intricacies and performance to user acceptance [8, 13, 14]. As a result, traditional approaches to AI integration often face obstacles such as long development cycles, high implementation costs, and difficulty in capturing real-world user interactions. Additionally, concerns about transparency, accountability, and bias in AI-driven systems further complicate the integration process.

In this paper, we explore integrating AI into interactive systems by utilizing a Wizard of Oz (WOz) approach to remediate and facilitate these challenges, as indicated by Browne [3]. Drawing inspiration from the classic WOz approach of simulating system capabilities through human intervention [4], we implemented a WOz tool that leverages both basic levels of technical automation (e.g., communication among system components, capturing and replaying input streams, logging) and human cognition (e.g., natural language understanding and conversation, system context definition, pattern recognition) to address the complexities inherent in AI integration. NexOz integrates a nexus architecture, characterized by a back-end core where all communication and interactions between technical components converge, with Wizard of Oz techniques, providing a unified platform for diverse simulations across a wide range of devices.

One of the primary benefits of leveraging a WOz approach is its ability to facilitate rapid prototyping and experimentation. Given the fast-paced AI developments, agility and adaptability are paramount. By employing NexOz and human operators to simulate AI functionalities, developers can rapidly prototype, evaluate, and iterate on AI-driven features without the need for fully implemented AI systems. This enables quicker validation of concepts and refinement of functionalities, and ultimately accelerates the development cycle. Furthermore, the use of a WOz approach in a test or operational environment provides an invaluable opportunity to collect data logs of user interactions [6, 11], which can serve as a resource for training and refining AI algorithms, and capture real-world user behavior, preferences, and feedback.

The paper's contributions are twofold, with the first being the introduction of an approach for AI system development that employs an iterative process to gradually reduce dependence on human inputs. Additionally, the paper discusses the application of the approach by presenting some practical scenarios, showcasing the incremental integration of AI-based modules in the context of a research project on a digital assistant in the manufacturing industry.

2 Positioning in literature and development process

In literature, the WOz approach has found applications in various domains, including natural language interactions [20], human-robot interactions [16], automated vehicles [5], data analysis [8]. One of the main motivations for employing a WOz approach when integrating AI is that it offers the means to explore and develop systems without immediately committing to extensive engineering efforts [19, 20, 24]. Initially, designers determine possible design features and are then able to assess the feasibility through the simulated environment [12, 16, 19].

Throughout the development process, it is essential to collect data from user interactions [7, 17] to improve the system and even use it to train future implementations of the AI, forming a continuous feedback loop where the AI (once integrated) can learn from its own actions, adapt, and improve over time. By logging and annotating these interactions [15], designers can gain valuable insights into user preferences, potential issues and areas of improvement. This data-driven approach, leveraged by the intelligence of a human operator, enables continuous refinement and optimization, ultimately leading to a more accurate and responsive AI system.

As the project progresses, the output from the exploratory WOz sessions can be analysed to determine what future interactions with the AI system could be like [15], by gradually incorporating more sophisticated functionalities. This incremental approach allows for a natural evolution of the system, with each iteration building upon the results of the previous ones [16, 23, 24].

Eventually, the system evolves into a partially integrated state where AI algorithms start to take over certain tasks and responsibilities. This state is sometimes referred to as a Wizard *with* Oz system (i.e., a human-centric method that uses real technology in a simulated environment) [7, 16].

3 Incremental AI integration with NexOz

Many software projects share a similar base architecture, where clients communicate both internally and with external resources. Additionally, numerous low-level functionalities (e.g., logging for analysis, interaction support, event handling) have to be provided [2]. This is reflected in our NexOz's architecture (Figure 1), which revolves around a back-end nexus hub that acts as a modular platform, in which components can be flexibly integrated or removed. This hub also serves as a proxy layer for external services, abstracting their complexities. Client devices all connect to the back-end hub, which in turn is connected to a service bus for its communication with external components. The WOz component is linked in such a way that it can inconspicuously emulate the behavior of both internal and external components, and thus all kinds of application logic.

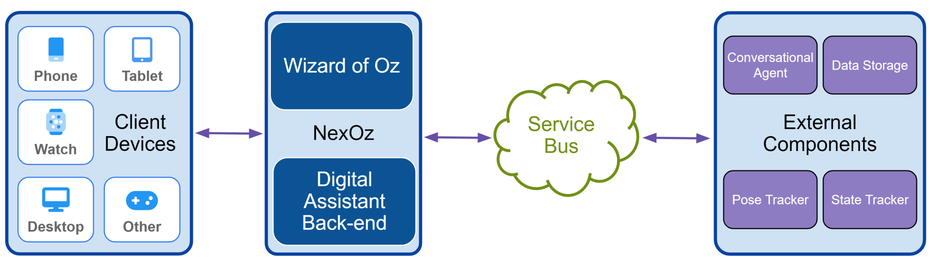


Fig. 1: Schematic overview of the architecture and positioning of NexOz (and a simplified, somewhat abstracted overview of the OperatorAssist project).

The architecture of NexOz centers around a SignalR hub, serving as the backbone for communication by allowing real-time, bidirectional Remote Procedure Calls (RPCs) between clients and the back-end hub. Additionally the back-end hub also connects to the Service Bus (through MQTT, REST, ...) to communicate with external components, such as the conversational agent. By proactively establishing the protocols for data exchange, by defining concrete payloads and message flows, the foundations for supporting specific AI functionality are laid. The WOz component then, in effect, defines an alternative path throughout the system, using the established protocols, to simulate AI behavior and responses.

To demonstrate our WOz approach, we discuss some practical scenarios in the context of the Flanders Make manufacturing industry research project 'OperatorAssist', a four-year project that started in July 2021. Due to the scale of the project and the complexity of the AI components, waiting for the development of those components would cause a significant bottleneck. NexOz was implemented to yield early results based on WOz simulations, which were incrementally replaced with actual AI components. Additionally, NexOz also enables flexible and repeatable tests. In the context of the OperatorAssist project, a test iteration often involves manually assembling physical components, during which the DA provides support based on, among others, a state tracker and conversational agent. NexOz can be used here to update the internal state to simulate an assembly process.

3.1 Specialized conversational agent

Using a conversational agent to provide answers to specific domain questions in an assembly environment poses several challenges. Firstly, the specialized domain knowledge presents a significant hurdle, as assembly tasks often involve intricate processes, technical terminology, and domain-specific jargon that may be unfamiliar to generic AI models. To limit user effort in manually posing (predictable) questions, the application will provide a convenient way to retrieve a predefined list of possible questions the user might have. Furthermore, the answer to specific questions can vary depending on the current context of the assembly (see Section 3.2).

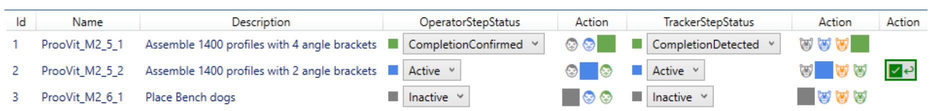
In NexOz, human operators can emulate responses in real-time, allowing developers to quickly test dialogue flows, interaction patterns, and conversation designs. Human operators can dynamically adjust their responses based on user inputs, mimicking the adaptability of conversational agents. NexOz allows sending predefined replies from a list of frequently asked questions and answers, while also allowing the human operator to provide custom answers. Consequently, NexOz's user-centric approach allows for quick adjustments and refinements based on user feedback, ensuring that the eventual conversational agent is intuitive, user-friendly, and aligned with user expectations.

The use of NexOz enabled us to incrementally build the DA, by mimicking the intelligence when there was no AI yet to respond to the user's prompts. An important part of this first phase is to define the payloads and message flows for the AI functionality, by constructing a concrete RPC protocol, which can still evolve with new insights. Later, actual AI functionality was added to the project to provide the user with a list of frequently asked questions to choose from, where the questions are filtered according to the current context. With this intermediate version of the AI component, it happens that the list of questions does not include the question that the user wants to ask, or that some part of the information is still missing. As the AI component does not yet offer a mechanism for handling free-text questions at this point, this must still be handled by the WOz. NexOz logs all these manual interventions, which gives the developers of the AI component insights into issues with its current implementation.

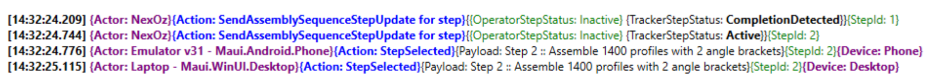
3.2 State tracking

The responses and behavior of the DA are based on an understanding of the physical context of the assembly, such as the current step in the assembly process or the tools and materials that are being used. This kind of contextual information is used, for instance, to filter questions by the conversational agent, as discussed in Section 3.1. Therefore, the system requires a robust tracking mechanism that establishes a representation of the physical state of the assembly, by relying on a set of AI techniques, including computer vision and multi-sensor fusion.

By simulating state tracking with a WOz, initial components of the project that are reliant on state information could be developed without an immediate need for a high-fidelity state tracker. NexOz, as illustrated in Figure 2a, provides the human operator with an orderly overview of the current assembly progress and facilitates easily updating or overriding of the perceived state. The overview shows both the step status as determined by the assembly operator and the status as it is perceived by the (simulated) state tracking module, where the status can easily be modified by clicking the appropriate action icon. Such modifications are communicated to the rest of the system through the established RPC protocols, rendering them indistinguishable from the eventual AI-based messages.



(a) State tracker module in NexOz.



(b) High-level logs after state update: Step 1 is detected as completed and Step 2 flagged as new active step. More details are logged and available for export.

Fig. 2: Screenshots of NexOz.

As the actual AI-based state tracking module evolved over subsequent iterations, the system became able to use computer vision to automatically determine the completion of steps, as well as the currently active step of the assembly. Despite these new capabilities, some actions remain hard to recognize reliably (e.g., actions with small or occluded parts) and still necessitate manual intervention within NexOz from time to time. All the manual interventions in NexOz are extensively logged (see Figure 2b for a summarized excerpt), whereby these logs serve as part of a two-way feedback loop to further optimize the AI-based module. In following iterations, we expect the state tracker to further optimize its recognition capabilities, diminishing the need for manual updates.

3.3 Gesture recognizer

The DA of the OperatorAssist project supports various input and output modalities, including different kinds of gestures, ranging from interactions with a Leap Motion controller [22] and smartwatch gestures [21] to unconstrained mid-air gestures [10]. The support for gestures is thus quite generic and open-ended.

One of the primary challenges with recognizing gestures lies in accurately capturing and interpreting nuances and variations. Gestures can vary significantly depending on factors such as cultural background, individual preferences, and environmental conditions. Therefore, an exploratory analysis is a valuable preliminary step: by systematically exploring different gesture options, soliciting user feedback, and iteratively refining gesture designs, developers can identify the most effective and user-friendly gestures for a specific task.

To support such an exploratory phase, we used a WOz approach to add gestures to our multimodal DA, similar to the approach of Salber and Coutaz [18]. NexOz provides the necessary features to test gestures without an actual AI recognizer, as a human operator can easily recognize gestures and trigger the associated interaction events through a configurable list of interactions (based on the use case at hand). Triggering such an event invokes a message that follows the established RPC protocol for gesture detection. NexOz also allows for logging the raw gesture data (e.g., finger movements detected by the Leap Motion controller) combined with the triggered event, resulting in labelled data. Once an AI recognizer is operational, NexOz continues to play a valuable role in logging, investigating and resolving errors in the gesture recognition.

4 Conclusion

This paper presents NexOz, our approach to incrementally integrate AI components into interactive systems using a Wizard of Oz. We demonstrated the practical application of this approach by showcasing several AI components, namely a specialized conversational agent, state tracker, and gesture recognizer. Overall, we argue that the use of NexOz offers a pragmatic solution to the complexity of integrating AI into interactive systems, providing a bridge between technological capabilities and user expectations. Through rapid prototyping and leveraging collected data logs, this approach empowers developers to navigate the complexities of AI integration.

Acknowledgments. This research is funded by Flanders Make under the project OperatorAssist_SBO, project number 2021-0133. Flanders Make is the Flemish strategic research center for the manufacturing industry in Belgium. This research also received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.

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

References

1. Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., Teevan, J., Kikin-Gil, R., Horvitz, E.: Guidelines for Human-AI Interaction. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–13. CHI '19, ACM (2019). https://doi.org/10.1145/3290605.3300233
2. Barros-Justo, J.L., Benitti, F.B., Matalonga, S.: Trends in Software Reuse Research: A Tertiary Study. Computer Standards and Interfaces **66** (2019). https://doi.org/10.1016/j.csi.2019.04.011
3. Browne, J.T.: Wizard of Oz Prototyping for Machine Learning Experiences. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–6. CHI EA '19, ACM (2019). https://doi.org/10.1145/3290607.3312877
4. Dahlbäck, N., Jönsson, A., Ahrenberg, L.: Wizard of Oz Studies — Why and How. Know.-Based Syst. **6**(4), 258–266 (1993). https://doi.org/10.1016/0950-7051(93)90017-N
5. Detjen, H., Pfleging, B., Schneegass, S.: A Wizard of Oz Field Study to Understand Non-Driving-Related Activities, Trust, and Acceptance of Automated Vehicles. In: 12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications. p. 19–29. AutomotiveUI '20, ACM (2020). https://doi.org/10.1145/3409120.3410662
6. Garcia, F.J.C., Lopes, J., Liu, X., Hastie, H.: CRWIZ: A Framework for Crowdsourcing Real-Time Wizard-of-Oz Dialogues. arXiv 2003.05995 (2020)
7. Green, A., Huttenrauch, H., Eklundh, K.: Applying the Wizard-of-Oz Framework to Cooperative Service Discovery and Configuration. In: RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759). pp. 575–580 (2004). https://doi.org/10.1109/ROMAN.2004.1374824
8. Gu, K., Grunde-McLaughlin, M., McNutt, A.M., Heer, J., Althoff, T.: How Do Data Analysts Respond to AI Assistance? A Wizard-of-Oz Study. arXiv 2309.10108 (2024)
9. Horvitz, E.: Principles of Mixed-Initiative User Interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 159–166. CHI '99, Association for Computing Machinery (1999). https://doi.org/10.1145/302979.303030
10. Hou, W., Feng, G., Cheng, Y.: A Fuzzy Interaction Scheme of Mid-Air Gesture Elicitation. Journal of Visual Communication and Image Representation **64** (2019). https://doi.org/10.1016/j.jvcir.2019.102637
11. Katsakioris, M.M., Hastie, H.F., Konstas, I., Laskov, A.: Corpus of Multimodal Interaction for Collaborative Planning. Proceedings of the Combined Workshop on Spatial Language Understanding (2019), https://api.semanticscholar.org/CorpusID:199553531
12. Kerly, A., Bull, S.: The Potential for Chatbots in Negotiated Learner Modelling: A Wizard-of-Oz Study. In: International Conference on Intelligent Tutoring Systems. pp. 443–452. Springer (2006)
13. Kocielnik, R., Amershi, S., Bennett, P.N.: Will You Accept an Imperfect AI? Exploring Designs for Adjusting End-user Expectations of AI Systems. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–14. CHI '19, ACM (2019). https://doi.org/10.1145/3290605.3300641

14. Luger, E., Sellen, A.: "Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. p. 5286–5297. CHI '16, ACM (2016). https://doi.org/10.1145/2858036.2858288
15. Okamoto, M., Yang, Y., Ishida, T.: Wizard of Oz Method for Learning Dialog Agents. In: Klusch, M., Zambonelli, F. (eds.) Cooperative Information Agents V. pp. 20–25. Springer Berlin Heidelberg (2001)
16. Riek, L.D.: Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. J. Hum.-Robot Interact. **1**(1), 119–136 (2012). https://doi.org/10.5898/JHRI.1.1.Riek
17. Rieser, V., Lemon, O.: Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz Data: Bootstrapping and Evaluation. In: Moore, J.D., Teufel, S., Allan, J., Furui, S. (eds.) Proceedings of ACL-08: HLT. pp. 638–646. Association for Computational Linguistics (Jun 2008), https://aclanthology.org/P08-1073
18. Salber, D., Coutaz, J.: Applying the Wizard of Oz Technique to the Study of Multimodal Systems. In: Bass, L.J., Gornostaev, J., Unger, C. (eds.) Human-Computer Interaction. pp. 219–230. Springer Berlin Heidelberg (1993)
19. Schlögl, S., Doherty, G., Karamanis, N., Luz, S.: WebWOZ: A Wizard of Oz Prototyping Framework. In: Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems. p. 109–114. EICS '10, ACM (2010). https://doi.org/10.1145/1822018.1822035
20. Schlögl, S., Doherty, G., Luz, S.: Wizard of Oz Experimentation for Language Technology Applications: Challenges and Tools. Interacting with Computers **27**(6), 592–615 (2015). https://doi.org/10.1093/iwc/iwu016
21. Shen, C., Ho, B.J., Srivastava, M.: MiLift: Efficient Smartwatch-Based Workout Tracking Using Automatic Segmentation. IEEE Transactions on Mobile Computing **17**(7), 1609–1622 (2018). https://doi.org/10.1109/TMC.2017.2775641
22. Ultraleap: Leap Motion Controller 2. https://leap2.ultraleap.com/leap-motion-controller-2/, [Accessed 03-05-2024]
23. Xie, T.: The Synergy of Human and Artificial Intelligence in Software Engineering. 2013 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE) pp. 4–6 (2013), https://api.semanticscholar.org/CorpusID:14096820
24. Xin, D., Ma, L., Liu, J., Macke, S., Song, S., Parameswaran, A.: Accelerating Human-in-the-loop Machine Learning: Challenges and Opportunities. In: Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning. DEEM'18, ACM (2018). https://doi.org/10.1145/3209889.3209897