



**HAL**  
open science

# Prototyping an intelligent satellite supervision system using Wizard of Oz

Yiran Zhang, Charlie Madier, Cédric Bach, Guillaume Calvet

► **To cite this version:**

Yiran Zhang, Charlie Madier, Cédric Bach, Guillaume Calvet. Prototyping an intelligent satellite supervision system using Wizard of Oz. IHM'23 - 34e Conférence Internationale Francophone sur l'Interaction Humain-Machine, AFIHM; Université de Technologie de Troyes, Apr 2023, Troyes, France. hal-04046116

**HAL Id: hal-04046116**

**<https://hal.science/hal-04046116v1>**

Submitted on 29 Mar 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Prototyping an intelligent satellite supervision system using Wizard of Oz

Prototypage d'un système intelligent de supervision de satellites à l'aide du Magicien d'Oz

YIRAN ZHANG, Human Design Group, France

CHARLIE MADIER, Human Design Group, France

CÉDRIC BACH, Human Design Group, France

GUILLAUME CALVET, Human Design Group, France

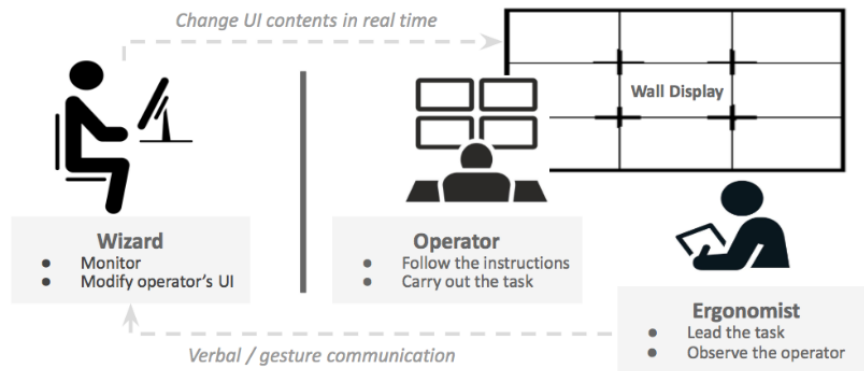


Fig. 1. WOz system for evaluating AI-driven satellite supervision systems in early prototype phase

Getting feedback from real users in the early prototyping stage plays an important role in risk management for the development of an AI satellite supervision system. In this work, we apply the WOz approach to simulate the role of AI in the system, which allows the application designer to evaluate the prototype before the development is finished. In this paper, we first detail three different use cases and explain how the WOz can be used to validate user workflow and address specific evaluation needs in the AI system design. In addition, we propose a web-based platform to enable the generic use of WOz in different scenarios. Our platform allows for remote monitoring and control of the user interface. More importantly, instead of hard-coding each use case independently, our platform is reusable in different scenarios by allowing the application designer to easily set up the required configurations.

CCS Concepts: • **Software and its engineering** → **Use cases; Software prototyping; Process validation.**

Additional Key Words and Phrases: Wizard-of-Oz, AI system design

Obtenir un retour de la part d'utilisateurs réels au début de la phase de prototypage joue un rôle important dans la gestion des risques pour le développement d'un système de supervision de satellites basé sur l'IA. Dans ce travail, nous utilisons l'approche WOz pour simuler le rôle de l'IA dans le système, ce qui permet au concepteur de l'application d'évaluer le prototype avant que le développement ne soit complètement terminé. Nous détaillons d'abord trois cas d'utilisation différents et expliquons comment le WOz peut être utilisé pour valider le flux de travail des utilisateurs et répondre à des besoins d'évaluation spécifiques dans la conception de systèmes d'IA. En outre, nous proposons une plateforme web pour permettre l'utilisation générique de WOz dans différents scénarios. Notre plateforme permet de surveiller et de contrôler à distance l'interface utilisateur. De plus, au lieu de coder en dur chaque cas d'utilisation

indépendamment, notre plateforme est réutilisable dans différents scénarios en permettant au concepteur de l'application de mettre en place les configurations requises.

Mots-clés additionnels : Magicien d'Oz, Conception du système d'IA

**Reference:**

Yiran Zhang, Charlie Madier, Cédric Bach, and Guillaume Calvet. 2023. Prototyping an intelligent satellite supervision system using Wizard of Oz. IHM'23 : Actes étendus de la 34ème conférence Francophone sur l'Interaction Humain-Machine, April 03–06, 2023, Troyes, France.

## 1 INTRODUCTION

In a satellite supervision system, the terrestrial control room receives data from the satellite, produces satellite images and other products, and makes them available to customers. To standardize the main interfaces within EO ground systems, the Domino-X project [5] was launched in late 2021. One of the goals of the project is to employ AI technology to assist satellite operators in their daily work and reduce the number of operators needed in the control room. However, since AI system development can be both costly and time-consuming, it is therefore important to validate the corresponding user interface (UI) and interaction before starting the actual development phase.

This work was carried out within the scope of the Domino-X project (WP4.7), which is a collaboration between Human Design Group, Airbus Defense and Space, and Capgemini. Its main objective is to use the Wizard-of-Oz (WOz) approach to simulate the AI role embedded in the satellite supervision system. As a result, the designers can iterate on the behavior of the AI and its associated UI to validate its functionality in the early prototyping phases. Gathering in-context user feedback before the AI system is fully developed plays an essential role in risk management and cost control during the product design life-cycle. The design of the AI system, including its virtual representation and parameter settings, can be refined according to the user's actual needs to make it more user-friendly. As a result, users will have more trust in the AI system and be more willing to use it. In addition, early-stage prototype testing allows application designers to validate user flows and filter out unnecessary features before development to reduce costs.

Applying the WOz approach to a complex AI system can be challenging as it typically involves many use cases and scenarios. In addition, a real-time UI monitoring and control system must be in place to allow application designers to observe user behavior and modify the UI in time. To meet these requirements, we propose a web-based platform that allows application designers to control the UI and modify its content remotely. Besides, instead of hard-coding each use case individually, our platform is reusable and provides application designers the flexibility to configure the parameters needed in different scenarios.

In the following sections, we first review the applications of the WOz approach in the literature and propose a design space to broadly classify them. Next, we detail three use cases of how the WOz method can be used to improve the envisioned AI-driven satellite supervision system. Finally, we present the design and development of the platform for the generic use of the WOz method in such prototype testing.

## 2 STATE OF ART

The WOz approach was first used by Michael Kelley in the 1970s to test conversational agent prototypes [6]. In this research, the user interacts with a conversational agent managed by a human operator working behind the scenes. The human operator, also known as the "wizard", ensures the system responds appropriately to the user's input. As a result, the user feels as if they are interacting with a fully functional prototype, even if the system is only partially

implemented. As the human "wizard" fills in the gaps in functionality, the WOz method allows application designers to test and iterate the interface design and interaction in the early design stages before building fully functional and expensive applications. Therefore, such an approach has been widely used lately in many application design processes for its low cost and in-context testing ability.

The design space of the WOz application can be broadly classified into four dimensions. First of all, depending on the degree of control that the "wizard" has over the prototype, WOz applications can range from a primitive mockup relying entirely on human manipulation to a fully functional application without additional human assistance. For example, paper prototyping is heavily based on the WOz technique. Unlike mature applications that respond to user input on their own, a human operator is necessary to manually adjust the prototype to the user's actions in paper prototyping [9].

The WOz applications can also be categorized based on the geographic location of the human operator and the user, i.e., whether the "wizard" and the user are in the same location during the evaluation. For example, paper prototyping usually requires a human operator to stay with the user at the same location to manually manipulate the prototype. The need for the co-localization of the user and operator limits the use of paper prototyping at the laboratory level. To extend paper prototyping to a wider range of uses, Google Glass can be used as a medium to exchange the live streams of the distributed user's first-person perspectives and the live streams of the paper prototypes to enable remote paper prototyping [4].

Depending on the design of the application, the degree to which users are aware that they are interacting with an operator-driven interface is different. This leads to a third dimension of the WOz application classification. User awareness can be affected by different factors, such as the degree of control of the "wizard", the geographical distribution of the operator and the user, and the latency of the system. In some cases, such as paper prototyping, the test user is fully aware that the system is being driven by a human operator.

Finally, WOz applications can be grouped by the scenarios in which they are used, from 2D mobile applications to simulated 3D environments. The WOz method is also used to capture requirements in the early stages of mobile application design [1]. To leverage WOz in the design of complex mobile applications, some digital prototyping tools, such as Lake [7], are offered to application designers to enable them to prototype with much higher fidelity. Besides, WOz is also used for some more advanced interfaces, including those that incorporate machine learning experiences [2] and those that use speech commands [10, 11]. In addition to traditional 2D interfaces, the WOz setup is also widely used for the evaluation of simulated 3D environments, such as analyzing users' natural hand interaction in an augmented reality interface [10], navigating the user in a virtual environment [12, 13] or evaluating an emotional agent integrated into a virtual environment [8].

Our work focuses on the application of the WOz approach to an AI-driven satellite supervision system, where the 2D operator interface is fully controlled by a human operator behind the scenes.

### 3 WOZ USE CASES IN DOMINO-X

This section details the possible use cases of the WOz system in three different scenarios identified in the Domino-X project. In these scenarios, we envision an AI-driven system that helps operators solve problems, anticipates potential issues, and proposes assistance to routine tasks based on their operational planning. The use of the WOz approach allows the application designers to refine the system using feedback from real users before the actual implementation of AI.

During the evaluation of the prototype, the application designers use the WOz system to observe user interaction and modify accordingly the operators' interface in real-time. The modifications can be triggered based on operator actions or specific evaluation needs. The former allows application designers to validate, for example, the user's workflow based on a sequence of tasks previously defined for the evaluation. The latter is more appropriate for modifying values based on the observation and needs of the application designer during evaluation, such as setting specific values to compare different user interface designs. Different aspects of the human factor, such as the cognitive load of the operator, the decision-making efficiency, and the error rate, can be taken into account for the evaluation as they have a large impact on the design choices.

### 3.1 AI Problem Solving

In the first scenario, the AI-driven virtual assistant suggests a list of possible solutions to help the operator solve the encountered problem. The proposed solutions are based on existing expert knowledge of system recovery procedures and on the learning of the operator's previous operations in solving similar problems. Each solution in the list provides the operator with a reliability level of the solution, a step-by-step procedure, and a description including relevant information about the problem. Operators can select a solution from the list. The selected solution can be performed automatically by the system with the permission of the operators, i.e., a high-level delegation at function level management through consent in the Letter of Agreement (LoA). Or it can be performed step by step by the system but under the supervision of the operators, i.e., a step-by-step delegation at the individual task level in the LoA. Otherwise, the operators can perform the solution manually with the assistance of the AI. The operators can also reject all suggestions after reading them and implement their own solutions.

To help operators gain a deeper understanding of the problem and make a final decision effectively, the operators must be provided with a reasonable explanation for each solution. In addition, the number of recommended solutions should also be limited to an appropriate value to control the amount of time they spend reading and understanding the solution before a final decision is made.

The WOz approach can help application designers to verify user interaction and gain a better insight into relevant variables before the system is developed. With the WOz method, the application designer can:

- Validate user workflow during the problem-solving process with different levels of automation.
- Modify the content of the solution at runtime to ensure that operators are provided with the appropriate information to make an efficient decision.
- Change the reliability level on the fly to verify how the quality of the solution proposed by the AI affects the operator's decision.
- Add/remove items from the solution list to determine an appropriate number of solutions to offer to operators.

### 3.2 AI Problem Anticipation

The second scenario focuses on the daily monitoring of the satellite system, especially on the prediction of potential problems in the near future. The system tracks various system parameters in real-time, such as CPU overhead and cloud node occupancy. Based on this data, the AI should use algorithms to predict the future evolution of these parameters and warn operators when these values are at risk of exceeding the alarm thresholds. The operator can then check the details of the warning and decide whether to prevent the risk or ignore the warning.

The UI design for such warnings is quite challenging, as it needs to provide enough awareness to the operator of the upcoming problem, while not interrupting the operator's ongoing task. In addition, the thresholds for triggering warnings for the anticipated problems must be carefully set. The inappropriate setting of the thresholds could severely disrupt the operators' daily work. For example, the system may be too sensitive to respond to the anticipated problems and therefore send a large number of warnings to operators. However, seeing a successive number of warnings in a short period of time can hinder the operator's work and make them easily overloaded.

To better design the representation of warnings and to guide the AI parameter settings, application designers can use the WOz method before the full development of the application to:

- Trigger warnings at different stages of the predetermined user flow to verify if the UI design can help the operator identify the problem detected by the AI.
- Obtain the maximum number of warnings that can be displayed to the operator at a given time, taking into account the operator's cognitive load.

### 3.3 AI Routine Support

In the third and final scenario, the AI suggests to the operator a daily routine based on a list of tasks scheduled at the beginning of each week. Possible suggestions include (i) adding a new task to the routine, e.g., adding an urgent customer request to the regular tasks; (ii) suspending a task in the routine, e.g., temporarily hanging some tasks due to system resource constraints detected by the AI; and (iii) reordering tasks in the routine, e.g., prioritizing some tasks based on the AI's learning of operator preferences and operation history. The operator can approve the new recommended routine, reject it, or customize their own routine based on the AI suggestions.

The manipulation of the routine list requires several small actions of the operator and is highly dependent on the design of the UI. Therefore, it is important to check in advance whether the UI design can adequately meet the operator's needs and to determine an appropriate time to remind the operator of the current routine change. The use of the WOz method can therefore be interesting in these cases since the application designers can:

- Send different propositions to the operator to verify that they can manipulate the list as expected.
- Trigger routine recommendations in different circumstances to determine the right time to provide new routines to the operator.

## 4 TECHNICAL SPECIFICATIONS

To meet the demands of the above-mentioned scenarios, three roles are envisioned in the WOz system. As illustrated in figure 1, a satellite operator (real user) manipulates the targeted interface and carries out a task following a predefined instruction. An ergonomist guides the operator through tasks and observes his/her behavior and interaction with the interface. The "wizard" (IT engineer) controls the operator's UI behind the scenes based on the operator's actions, making him/her believe that he/she is interacting with an AI-driven virtual assistant.

To allow the "wizard" to monitor the operator's actions and modify their UI on the fly, a remote monitoring and control platform for the generic use of the WOz method is designed. The platform is developed with Java Framework (Quarkus). It includes three main functions: remote monitoring, UI variable control, and UI state control, which are detailed in the following sections.

#### 4.1 Global view for remote monitoring

The global view in the platform displays a live stream of the operator's screen allowing the "wizard" to track the operator's action. To do this, a python script (`sender.py`) is executed on the PC that the "wizard" needs to monitor. The script takes a screenshot of the operator's desktop, frame by frame, and sends it to the WOz platform via web socket. The platform then displays the frames when receiving them.

In case the operator works with a wall-sized display, the platform also supports multiple screens monitoring (see figure 2). The "wizard" can switch the global view from one screen to another by clicking on the targeted screen. To display multiple screens, a second python script (`receiver.py`) should be executed on the "wizard" side. It receives streaming data from multiple screens (`sender.py`) and sends the data to the WOz platform (see figure 3). To synchronize multiple screens, the message must be reorganized before being sent to the WOz platform and it keeps the following format: `[nb frames] [frame_1 size] ... [frame_N size] [frame_1 data] ... [frame_N data]`.

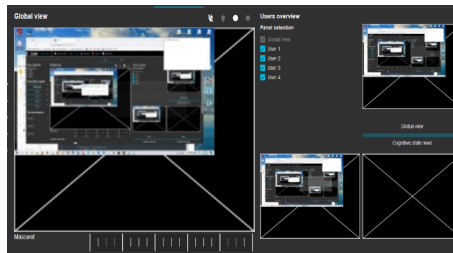


Fig. 2. Multiple screen monitoring

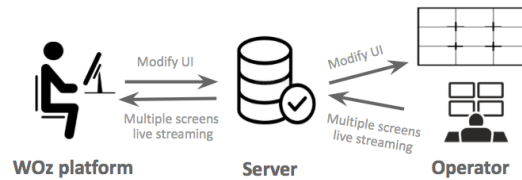


Fig. 3. Communication architecture of the WOz system

#### 4.2 Variables controls panel

The variable control panel allows the "wizard" to modify specific variables in the operator interface to meet specific evaluation needs, such as the content and reliability level of the solution in the AI problem-solving use case (see 3.1). The "wizard" can set the message and send it to the operator interface by clicking the "submit" button. After receiving the message on port 8080, the UI modifies its content accordingly.

The variables of the control panel must be defined beforehand by the application designers according to their test needs. As different variables can be used in different scenarios, a quick configuration of the control panel in the WOz platform is thus necessary. To help application designers configure the panel automatically instead of hard-coding it, our platform allows application designers to select one of the configuration files in the dialog by clicking the "open file" button (see figure 4).

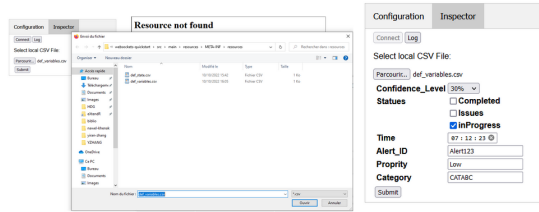


Fig. 4. Variable control panel automatically generated according to the selected configuration file

The configuration file includes a set of predefined variables. Four types of variables are supported by our platform: the text value, the time value, the drop-down list, and the checkbox. The application designer can define the values in the configuration file by specifying its name, type, and default values. The platform will automatically add them to the panel using JavaScript. For example, to define a time value named Time and whose default setting is 07:12:23, the configuration file should be defined as "Time time 07:12:23" (see more examples in figure 5).

```

1 Category text CATABC
2 Property text Low
3 Alert_ID text Alert123
4 Time time 07:12:23
5 Statuses checkbox inProgress checked Issues not_checked Completed not_checked
6 Confidence_Level dropdown 30% 50% 70% 100%
    
```

Fig. 5. Example of the configuration file for generating the variable control panel in the platform

### 4.3 States control panel

The state control panel is used to allow the "wizard" to modify the operator interface according to the operator's actions. Unlike modifying some specific variables in the UI content, the change of state triggers a series of predefined actions in the operator interface. For example, in the AI system monitoring use case (see section 3.2), sending a warning to the operator interface may lead to changes in the AI representation, trigger some alarm sounds, and display a pop-up menu, among other things. Different states should be defined in advance according to the target user flow. Figure 6 shows an example of predefined states and associated transitions for monitoring and maintenance of the satellite system.

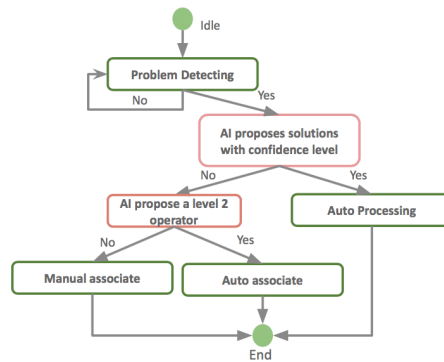


Fig. 6. States predefined for satellite system monitoring and maintenance



To facilitate the configuration of the WOz platform, our platform allows application designers to select a predefined configuration and set up panels automatically. The configuration of the state keeps a table format, each row should contain the current state of the system, the operator's possible action, and the leading next state (see figure 7).

Current state	Input	Next state
Detect problem	No	Detect problem
Detect problem	Yes	AI propose Solutions with CI
AI propose Solutions with CI	No	AI propose a level 2 operator
AI propose Solutions with CI	Yes	Auto processing (end)
AI propose a level 2 operator	No	Manual associate (end)
AI propose a level 2 operator	Yes	Auto associate (end)

Fig. 7. Example of the configuration file for the state control panel

The panel starts with an idle state. The possible actions of the operator are automatically displayed in a drop-down list with the resulting following step indicated next to it. The application designer can then select one of the operator actions from the list and click the "submit" button to update the operator interface (see figure 8). The action is sent with a string format using port 8080. The operator interface then identifies the states on receipt of the message and executes a series of predefined functions associated with the action.

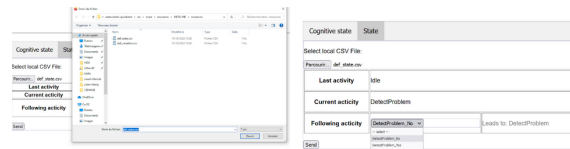


Fig. 8. State control panel

## 5 CONCLUSIONS AND PERSPECTIVES

The embedding of AI technologies into a satellite supervision system can be time-consuming and resource-intensive. It is therefore important to obtain real user feedback to validate the interface design before system development begins. In this paper, we present three detailed use cases focusing on how the WOz approach can help application designers evaluate an AI-driven satellite supervision system in the early prototyping phase. The main idea is that the application designers remotely observe the user's action and modify the UI content accordingly to simulate the role of the AI. The evaluation can be performed according to a predetermined sequence of tasks or be driven by the specific needs of the application designer. During the assessment, different aspects of human factors, such as user decision-making efficiency, can be taken into account. As a result, application designers can gather user feedback and refine the system design before system development is fully complete. Besides, we propose a web-based platform dedicated to helping application designers quickly set up the remote monitoring and UI control for the generic use of the WOz approach. This platform can be integrated in the future with other existing platforms, such as Ngager[3], which will allow the WOz methods to be extended from 2D interfaces to 3D environments.

## REFERENCES

- [1] Zahra Shakeri Hossein Abad, Shane DV Sims, Abdullah Cheema, Montasir B Nasir, and Payal Harisinghani. 2017. Learn more, pay less! lessons learned from applying the wizard-of-oz technique for exploring mobile app requirements. In *2017 IEEE 25th international requirements engineering conference workshops (REW)*. IEEE, 132–138.

- [2] Jacob T Browne. 2019. Wizard of oz prototyping for machine learning experiences. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.
- [3] G. Calvet, G. Jegou and D. Lewkowicz. 2020. Ngagement, une plateforme de design et d'évaluation rapide. In *AActes du 55ème Congrès de la SELF*. 16–18.
- [4] Kevin Chen and Haoqi Zhang. 2015. Remote paper prototype testing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 77–80.
- [5] Novaka Daniel, Marcell, Annaneb Amina, Baillarde Régis, Berryd Alain, Brandone Cédric, Desormeau Vincent, Gaudang Sylvain, Joyeuxh Julien, Kaethleri Stan, Madier Charlie, Meletk Olivier, and Rouxl Yann. 2022. Future ground segments with standardized interfaces: the DOMINO-X project. In *73rd International Astronautical Congress (IAC), Paris, France*.
- [6] Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. 2005. Wizard of Oz support throughout an iterative design process. *IEEE Pervasive Computing* 4, 4 (2005), 18–26.
- [7] Andrew Finke. 2019. Lake: A digital wizard of oz prototyping tool. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.
- [8] Károly Hercegf, Anita Komlódi, Máté Köles, and Sarolta Tóvölgyi. 2019. Eye-tracking-based wizard-of-oz usability evaluation of an emotional display agent integrated to a virtual environment. *Acta Polytechnica Hungarica* 16, 2 (2019), 145–162.
- [9] John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)* 2, 1 (1984), 26–41.
- [10] Minkyung Lee and Mark Billinghurst. 2008. A wizard of oz study for an ar multimodal interface. In *Proceedings of the 10th international conference on Multimodal interfaces*. 249–256.
- [11] Martin Porcheron, Joel E Fischer, and Michel Valstar. 2020. NottReal: A tool for voice-based Wizard of Oz studies. In *Proceedings of the 2nd Conference on Conversational User Interfaces*. 1–3.
- [12] Nhan Tran, Josh Rands, and Tom Williams. 2018. A hands-free virtual-reality teleoperation interface for wizard-of-oz control. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*.
- [13] Xin Zou, Steve O'Hern, Barrett Ens, Selby Coxon, Pascal Mater, Raymond Chow, Michael Neylan, and Hai L Vu. 2021. On-road virtual reality autonomous vehicle (VRVAV) simulator: An empirical study on user experience. *Transportation Research Part C: Emerging Technologies* 126 (2021), 103090.