A Physical User Interface Description Language for Interface Digital Twins

Nicholas Caporusso[1], Aryan Tandon[1],

Tewodros Amare[1], Sarah Neace[1], Samuel Sungmin Cho[1]

[1] Department of Computer Science, Northern Kentucky University,
Louie B Nunn Dr, 41099 Highland Heights, United States
{caporusson1, tandona1, amaret1, neaces3, chos5}@nku.edu

ABSTRACT

Recent studies demonstrated that the user interfaces of many physical products lack accessibility to individuals with disabilities, with specific regard to individuals who are blind or suffer from cognitive conditions. Nevertheless, in the last decades, the introduction of User Interface Description Languages (UIDLs) in the software industry has provided designers and developers with a set of standard protocols for formalizing the components of the User Interface (UI) of applications and websites and for specifying their properties. This, in turn, has resulted in best practices that enable more rapid development of accessible UIs. In our previous work, we introduced the concept of Interface Digital Twins (IDTs), that is, augmented digital replicas of the user interfaces of physical devices. In this paper, we present a UIDL specifically designed for improving the accessibility of physical products: we detail the language, highlight its advantages, and discuss its application in the context of IDTs.

Keywords: Human-Machine Interaction, accessibility, universal access.

INTRODUCTION

Although the American Disability Act defines guidelines and regulations in terms of equitable access for digital and physical products, the accessibility divide between hardware devices and software technology is increasing. In the last decade, accessibility has received increasing attention in software and, particularly, in the design of the user interface of websites and desktop and mobile applications, where a set of standards and guidelines have been created and incorporated in development tools (Ekstrand, 2017) (Acosta-Vargas et al., 2019). On the contrary, the design of physical products such as home appliances, electronic devices, and vending machines, lacks accessibility standards and requirements (Lee et al., 2018). As a result, many products ignore or fail to adopt the most basic accessibility rules, which impacts the quality of life of people with disabilities (Caporusso et al., 2014) (Caporusso et al., 2019). This, in turn, specifically impacts individuals who are blind or suffer from cognitive conditions, who are unable to independently learn and use several physical products and devices due to the lack of accessibility of their User Interface (UI). Moreover, prior research on vending machines demonstrated that the introduction of novel interface components that do not incorporate any tactile cues (e.g., touchscreens) significantly impacts accessibility, making it nearly impossible to complete a purchase without assistance (Caporusso et al., 2019). Our previous work (Caporusso et al., 2020) (Caporusso et al., 2020) introduced the concept of Interface Digital Twin (IDT), that is, the digital replica of the UI of a physical product rendered in the form of an interactive model: IDTs leverage the features of the smartphone (e.g., text-to-speech, vibration, and the possibility of adding more information) to increase the accessibility of the interface of a physical device (e.g., translate visual labels into speech), help individuals understand its components and learn how to operate it (e.g., interactive user manual), and support users in accomplishing their tasks on the actual device (e.g., provide step-by-step guidance). The goal of IDTs is to improve the accessibility of the physical UI of currently available technology without requiring any modifications to the product, which makes them suitable for enhancing the accessibility of existing devices at no additional cost. In addition to supporting individuals with sensory or cognitive disabilities, IDT can be utilized to abstract and digitize the functionality of physical devices. For instance, in the case of a ticket vending machine, the entire process can be realized digitally, using a simplified interface for buying the ticket on the smartphone. IDTs require a User Interface Description Language (UIDL) that enables describing the components of the UI of physical products and their properties Fin a formal representation. Although several UIDLs have been developed in the last decades, they are not easily applicable to physical products due to the inherent diversity of the components of the UI of physical products and to the nature of the IDT-creation process, which mostly consists of reverse-engineering the UI. Therefore, in this paper, we detail a Physical User Interface Description Language (PUIDL) that has the purpose of representing the main characteristics of the UI of a physical device into a digital format so that it can be utilized to create IDTs on smartphones and other personal devices. We introduce the language, and we highlight the differences between the proposed PUIDL and existing User Interface Description Languages, which are designed to represent software interfaces, only. We detail the main characteristics of the language, we discuss its syntax and notation, and we demonstrate how the proposed PUIDL can be utilized to implement the IDT of an actual physical device. Furthermore, we analyze the effectiveness, modularity, interoperability, and performance of the proposed language in describing physical user interfaces. Finally, we describe how PUIDLs can support the use IDTs for controlling physical devices in the context of the Internet of Things (IoT).

RELATED WORK

UIDLs enable producing a formal description of the UI of a software application using a format that is independent of the programming language utilized in the implementation Shaer et al., 2008. As UIDLs specify what the components of an interface look like rather than how they are created, the representation of the UI (i.e., the presentation layer, or the view) is separated from the specification of the internal logic (i.e., the controller) of an application. Representing the UI of an application using a UIDL has several advantages which, in general, result in simpler development and deployment of software products. For instance, the formal specification of a UI can be given as an input to a system that automatically parses all the characteristics of the components and generates the interface. Furthermore, UIDLs enable achieving a standard protocol that can be utilized by multiple programming languages to render the same UI on different devices and environments (e.g., web, desktop, or smartphone applications). In the last decades, several UIDLs were introduced to represent the graphical user interface (GUI) of websites and applications. Most specifications use markup languages such as XML because of its modular structure, interoperability, and extensibility (Abrams et al., 1999). Although the simplicity of XML-based UIDLs resulted in a large adoption in the software industry, they are especially suitable for the development of cross-platform mobile applications (Mitrovic et al., 2016). Recently, ed the use of UIDLs has been extended to different types of applications in several contexts, from fintech (Ulusoy et al., 2019) to virtual reality (Cruz-Carrizales et al., 2021), whereas others have proposed UIDLs for describing the entire flow of user actions (Fatima et al., 2019) or multimodal interfaces (Gaouar et al., 2018), to cope with an increasingly richer and more complex scenario of interaction dynamics. Also, as proposed by (Khan & Khusro, 2021), accessibility can be incorporated as an aspect of the formal description of UIs to enhance the user experience of individuals who are blind. Several research groups have attempted to apply the concept of UIDLs to physical devices for achieving a standard that simultaneously enables to incorporate accessibility in the design of their UIs (Zuehlke et al., 2004) (Nichols & Myers, 2009) (Miñón et al., 2014). Unfortunately, currently, there is no ubiquitous or established protocol for supporting individuals who are blind in independently learning and using the most common physical devices.

A PHYSICAL USER INTERFACE DESCRIPTION LANGUAGE

The objective of our work is to create a PUIDL that can be utilized for describing the UI of existing physical products in a format that is suitable for IDTs (that is, from a user's standpoint), rather than accurately representing the structure of devices from a designer's perspective. Typically, the purpose of existing UIDLs is to specify how the interface will be rendered on a particular device (i.e., they are primarily prescriptive). In contrast, the proposed PUIDL has a more descriptive nature, because it translates the visual structure of an existing UIs into augmented models containing additional information that can be explored using multimodal labels and feedback (e.g., speech and vibration) so that individuals who are blind or suffer from cognitive impairments can be presented with a spoken description of the UI layout on their smartphone and receive assistance by the system while independently learning and using the different components of a physical device. IDTs utilize HTML as the preferred language for digitizing the interface and translating its visual components into text labels that can provide a high-level description of the entire interface as well as a more detailed explanation of the individual components. To this end, the proposed PUIDL also relies on a parser that enables translating the structural and functional aspects of the UI layout and controls into descriptions rendered as text or audio (i.e., speech). Figure 1 represents the typical usage scenario for an IDT: the user utilizes their smartphone to scan the tactile/visual tag (e.g., QR code) placed on a physical device; the tag encodes a URL that contains the digital version of the physical UI (i.e., an HTML page rendered based on the specification written in the proposed PUIDL); the IDT leverages the accessibility features of the smartphone to describe the layout and the components of the interface to the user with a text-to-speech engine, and provides them with additional guidance for learning how to use the device and accomplishing their tasks.

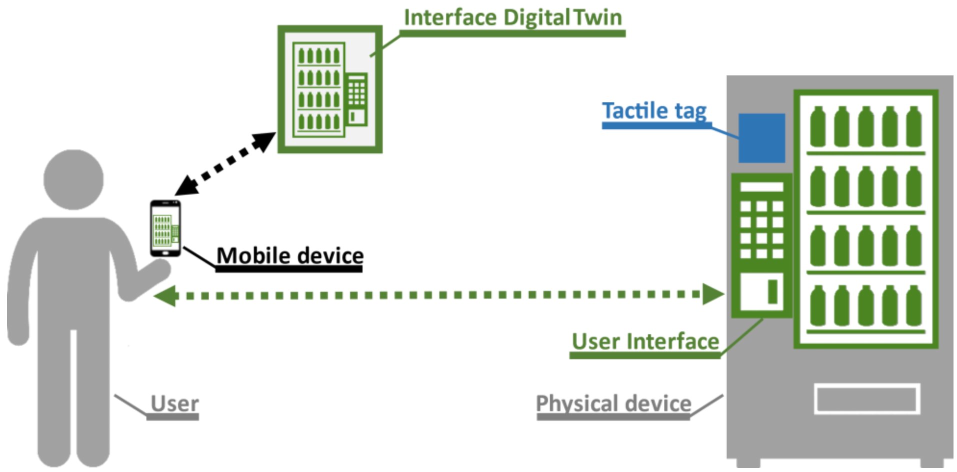

Figure 1. Overview of a usage scenario of IDTs with eVending (Lee et al., 2018).

Language Notation

The preferred implementation language for our PUIDL is JavaScript Object Notation (JSON). We preferred this format over other standards currently utilized for describing interfaces (e.g., XML) because its simplicity, speed, and interoperability make it an ideal replacement for older formats. As JSON is a schema-less representation, it has minimal overhead and supports evolving the language and adding new components without requiring any changes to any predefined formatting, which makes it particularly convenient for our PUIDL, which is still in a developing stage. As JSON is a text-based representation, it does not require any specific software or library to be opened. Furthermore, as the data is structured into key-value pairs, ordered lists, numbers, and strings, it is easy to read and understand even without the use of a dedicated parser and visualization tool, and it can easily be read and modified with a text editor. Although JSON originated from JavaScript, nowadays it is supported in most programming languages, either natively or through libraries. Moreover, JSON is commonly, but not exclusively, used to exchange information between web clients and web servers and over the last decades has become the format of choice for almost every web service on the web. This aspect renders the IDL description compatible with applications running on the web as well as other types of use involving the HyperText Transfer Protocol (HTTP) or other information exchange protocols. Indeed, JSON lacks the expressivity of a proper programming language, and it does not incorporate more complex elements such as functions. Nonetheless, this results in an advantage from an IDL standpoint because it forces users to use the IDL for describing the interface and its use, rather than programming additional features that could hinder the readability and interoperability of the IDT. The complete specification of the language and the HTML and text-to-speech parsers are available in the official repository of the project (Caporusso, 2021). Figure 2 demonstrates the process of modeling a physical UI into its corresponding IDT using the proposed PUIDL. Also, the figure shows the parsing process that translates the JSON of an IDT into an HTML layout or speech, using the JavaScript PUIDL parser.

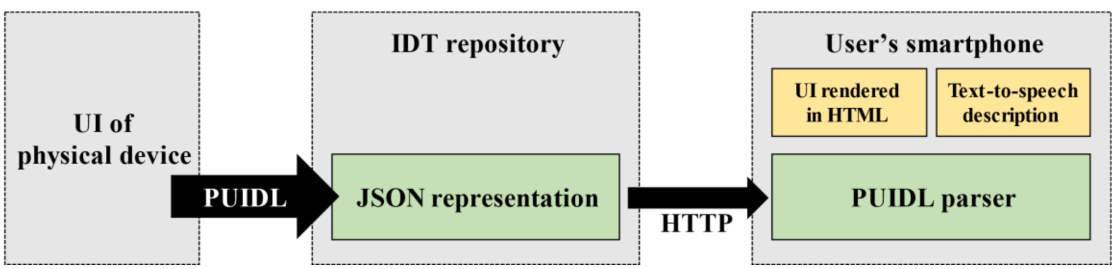

Figure 2. The process that translates the UI of a physical device into an IDT. The proposed PUIDL outputs a JSON file that can be translated into an HTML layout and speech using the system's PUIDL parser.

Interface Layout

The primary role of the proposed language is to offer a formal description of the structure of the interface and the characteristics of its components so that users can understand and use them. Typically, UIDLs represent the elements of the GUI of software applications, which are organized on two-dimensional spaces. Conversely, as physical devices consist of three-dimensional objects having different shapes, the components of their UIs are distributed on a more complex structure and consisting of multiple sides, which represents an additional challenge. Our PUIDL incorporates the concept of shape to describe the structure of an object: in addition to 2d and 3d primitives (e.g., rectangles, cubes, and spheres), our language supports multiple complex 3d shapes, which are implemented as polygons and associated with text descriptions. Regardless of their shape, the proposed language describes physical objects as a cube. By doing this, we represent them as having six sides (i.e., top, bottom, front, rear, left, and right), to make it easy to describe the positioning of the UI using simple and intuitive text labels. For instance, most home appliances such as microwave ovens have their UIs on the front side, whereas in other devices (e.g., washer, dryer, or stove) the UI controls can be located on multiple sides (e.g., top and front sides). In our language description, each side can contain zero or more panels, that is, an active area of the interface where UI components are located. Panels and their UI controls are represented as structured in rows, columns, or grids. By doing this, the proposed PUIDL can be utilized to convert the representation of the interface into natural language and speech that describes how many panels are in the UI and their absolute and relative location, using intuitive labels such as above, below, left, and right. The description of panels and sides supports attributes such as size and alignment, which enable dimensioning and positioning the elements in the rendered HTML output as well as in their spoken description. Figure 3 shows the interface of a microwave oven and the representation of its layout using the proposed PUIDL.

Figure 3. The UI of a microwave oven (left) and the digital representation of its layout using the proposed PUIDL (right), shown as a 3D rendering. The layout consists of a panel structured into three columns that contain the door (left), the handle (middle), and a panel (right). The latter column contains three rows: the first one contains the LED display, the second row contains a grid with push buttons for the pre-defined cooking options, and the third one contains a grid with push buttons for the numbers and the start and stop buttons.

UI controls

In addition to representing the layout of the interface of a device, the proposed PUIDL can be utilized to describe physical UI controls, including their type, positioning, purpose, and available options. The language defines the most common types of input (sensors) and output (actuators) devices, including buttons, switches, dials, displays, handles, doors, keypads, and other common controls. Each type is associated with a specific set of attributes that enable describing its form and function in detail (e.g., shape, color, interaction options). The availability of types makes it very convenient to reuse predefined components and quickly translate existing UIs into their corresponding IDTs. Nevertheless, the proposed PUIDL can be extended with new control types, which can be defined and referenced within the JSON object, as shown in Figure 4 and documented in the full specifications of the language. As the main purpose of the proposed PUIDL is to model the characteristics of the interface that is relevant for IDTs, the language captures a minimal set of attributes that enables understanding the purpose and operation of each control, whereas the details can be specified into a text description. UI components also support attributes such as size and alignment, which enable dimensioning and positioning the elements in the rendered HTML output.

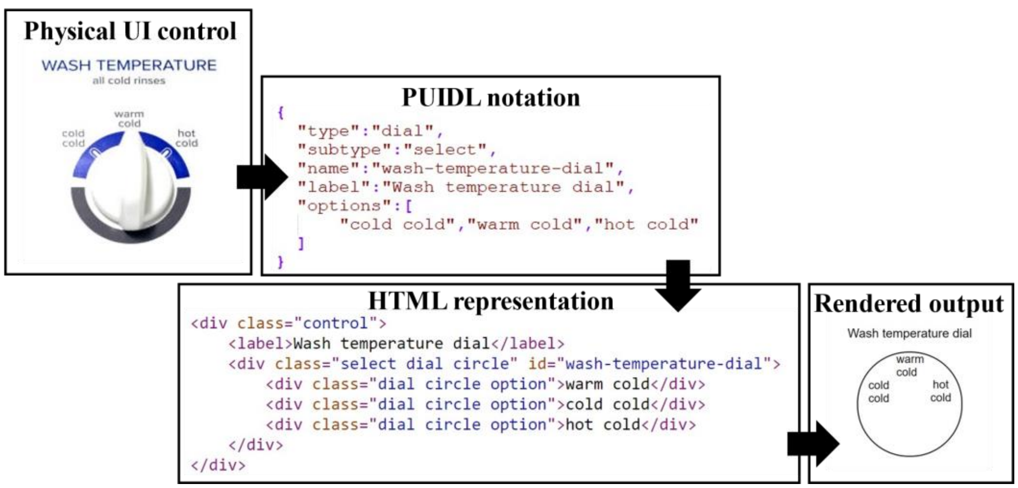

Figure 4. The modeling and rendering process of a UI control.

Additional details and product information

As IDTs are accessible digital replicas of the UI of existing devices and products, they can be retrieved via a web URL. Indeed, one of the objectives of IDTs is to facilitate their reuse. Thus, they can be associated with their physical counterparts using QR codes or other forms of interactive labels, so that all the instances of a specific product can share the same IDT. To this end, the proposed PUIDL supports incorporating additional details and product information such as make, model, year, version, and product codes (e.g., barcode or stock keeping unit). By entering this information, the author of an IDT can publish their PUIDL code (i.e., JSON) within the platform and make it searchable by others, who can directly use it. Also, designers who want to create a new IDT for a product can search for the IDTs of devices with similar interfaces, make the necessary changes, and publish their PUIDL code with the appropriate product information. By doing this, the language enables creating new IDTs and enriching the available devices mapped in the IDT ecosystem with very low effort. For instance, as many home appliances manufactured by different brands have similar interfaces, users can search for the closest IDT, adapt it to match the UI of their device, and upload the new product model in the IDT database. In addition to product information, which is mandatory, the PUIDL language supports specifying optional details about the product, such as color, size, and weight, which may be useful in some interaction scenarios.

CONCLUSIONS AND FUTURE WORK

Incorporating accessibility in existing physical devices is extremely difficult as it often requires making significant changes to products. Therefore, our work focuses on IDTs, that is, digital replicas of the interface of existing physical devices that translate and augment the visual components of UIs into speech and vibration, thanks to the accessibility features of the smartphone. The goal of IDTs is to support individuals who are blind or suffer from cognitive conditions in independently understanding the structure of the interface, learning its components, and using its functionalities to accomplish tpheir goals. In this paper, we introduced a language for achieving a formal description of the UI of physical devices and products. To this end, we proposed a PUIDL that is specifically designed for representing existing UIs and describing them to the users, rather than generating a formal model of the interface. In this paper, we introduced the first version of the language, described its purpose, and outlined its components. However, we focused on the salient aspects only, as we could not discuss all the implementation details due to space limitations and because the proposed PUIDL is still in its development stage. An updated and complete specification of the language is available in the official repository of the project (Caporusso, 2021).

REFERENCES

Ekstrand, V.S., 2017. Democratic governance, self-fulfillment and disability: Web accessibility under the Americans with disabilities act and the first amendment. Communication Law and Policy, 22(4), pp.427-457.

Acosta-Vargas, P., Salvador-Ullauri, L.A. and Luján-Mora, S., 2019. A heuristic method to evaluate web accessibility for users with low vision. IEEE Access, 7, pp.125634-125648.

Lee, J.H., You, N., Lee, J., Lee, K.J., Yun, M.H., Han, J., Jeong, Y.J. and Lee, H., 2018, August. Investigation of Accessibility Issues for Visually Impaired People When Using Washing Machines. In Congress of the International Ergonomics Association (pp. 1456-1464). Springer, Cham.

Caporusso, N., Trizio, M. and Perrone, G., 2014. Pervasive assistive technology for the deaf-blind need, emergency and assistance through the sense of touch. In Pervasive health (pp. 289-316). Springer, London.

Caporusso, N., Udenze, K., Imaji, A., Cui, Y., Li, Y. and Romeiser, S., 2019, July. Accessibility Evaluation of Automated Vending Machines. In International Conference on Applied Human Factors and Ergonomics (pp. 47-56). Springer, Cham.

Caporusso, N., Elleman, N., Cho, S.S. and Perilli, I., 2020, March. A Digital Platform for Improving Accessibility in Physical User Interfaces. In 2020 6th International Conference on Information Management (ICIM) (pp. 177-182). IEEE.

Caporusso, N., Elleman, N. and Cho, S.S., 2020. Interface Digital Twins: Rendering Physical Devices Accessible to People who are Blind. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 982-986). IEEE.

Shaer, O., Jacob, R.J., Green, M. and Luyten, K., 2008. User interface description languages for next generation user interfaces. In CHI'08 Extended Abstracts on Human Factors in Computing Systems (pp. 3949-3952).

Abrams, M., Phanouriou, C., Batongbacal, A.L., Williams, S.M. and Shuster, J.E., 1999. UIML: an appliance-independent XML user interface language. Computer networks, 31(11-16), pp.1695-1708.

Mitrovic, N., Bobed, C. and Mena, E., 2016, October. A review of user interface description languages for mobile applications. In the Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM.

Ulusoy, S., Batıoğlu, A. and Ovatman, T., 2019. Omni-script: Device independent user interface development for omni-channel fintech applications. Computer Standards & Interfaces, 64, pp.106-116.

Cruz-Carrizales, C.A., Bautista-Rojas, L.E. and Pedraza-Ferreira, G.R., 2021. Evaluation of User Interfaces Description Languages to Augmented Reality. Revista Facultad de Ingeniería, 30(57), pp.e13304-e13304.

Fatima, I., Anwar, M.W., Azam, F., Maqbool, B. and Tufail, H., 2019, October. Extending interaction flow modeling language (ifml) for android user interface components. In International Conference on Information and Software Technologies (pp. 76-89). Springer, Cham.

Gaouar, L., Benamar, A., Le Goaer, O. and Biennier, F., 2018. HCIDL: Human-computer interface description language for multi-target, multimodal, plastic user interfaces. Future Computing and Informatics Journal, 3(1), pp.110-130.

Khan, A. and Khusro, S., 2021. A mechanism for blind-friendly user interface adaptation of mobile apps: a case study for improving the user experience of the blind people. Journal of Ambient Intelligence and Humanized Computing, pp.1-31.

Zuehlke, D., Mukasa, K., Boedcher, A. and Reuther, A., 2004. useML–A Human-Machine Interface Description Language. " Developing User Interfaces with XML: Advances on User Interface Description Language.

Nichols, J. and Myers, B.A., 2009. Creating a lightweight user interface description language: An overview and analysis of the personal universal controller project. ACM Transactions on Computer-Human Interaction (TOCHI), 16(4), pp.1-37.

Miñón, R., Moreno, L., Martínez, P. and Abascal, J., 2014. An approach to the integration of accessibility requirements into a user interface development method. Science of Computer Programming, 86, pp.58-73.

Nicholas Caporusso Interface Digital Twins [Official repository] https://github.com/NicholasCaporusso/InterfaceDigitalTwins, 2021