A Digital Platform for Improving Accessibility in Physical User Interfaces

Nicholas Caporusso
Department of Computer Science
Northern Kentucky University
Highland Heights, United States
e-mail: caporusson1@nku.edu

Nicholas Elleman
Department of Computer Science
Northern Kentucky University
Highland Heights, United States
e-mail: ellemann1@nku.edu

Samuel Sungmin Cho
Department of Computer Science
Northern Kentucky University
Highland Heights, United States
e-mail: chos5@nku.edu

Ivan Perilli
University of Westminster
London,
United Kingdom
e-mail: ivan_perilli@yahoo.co.uk

***Abstract*— In the recent decades, accessibility received increasing attention. As a result, nowadays, established principles (e.g., Universal Design) contribute to creating products that take into consideration the diverse needs and abilities of all individuals. Unfortunately, there is a gap between the digital and physical world: while websites and user interfaces are adopting accessibility principles, most devices do not support an equitable use. Specifically, individuals who are blind are not able to independently learn and operate several categories of physical products. In this paper, we propose a system for rendering the User Interface (UI) of physical devices and products accessible to individuals with disabilities, with specific regard to people who are blind. To this end, we introduce the concept of Interface Digital Twin (IDT), that is, a digital replica of the physical UI that leverages the accessibility features of mobile devices (e.g., vibration and text-to-speech) to render the interface accessible, help individuals understand its components and learn how to operate it, and support users in accomplishing their tasks on the actual device. The advantage of the proposed solution is that it can be integrated in current technology without requiring any modification. We detail the architecture of the system and we discuss its advantages and use cases.**

Keywords- accessibility; universal design; user interface, Assistive Technology.

I. INTRODUCTION

Improving product usability and, particularly, reducing access barriers is crucial for serving an increasing number of users with diverse needs and abilities, such as the elderly and people with cognitive disorders or physical impairments. Supporting individuals who are blind is especially important, because their population is growing at a fast pace: according to recent statistics [1], it is expected to double and reach 2 million people by 2060, in the United States alone. This, in turn, urges for solutions that can be adopted immediately.

Indeed, the introduction of Universal Access principles resulted in improvements in several aspects of Information Technology (IT), over the last decade, and led to a more equitable use of devices and products. As an example, increased accessibility support in smartphones and tablets, the introduction of new features in web browsers (e.g., magnification), new HyperText Markup Language (HTML) components, such as semantic tags and attributes for Accessible Rich Internet Applications (ARIA), and enhancements to Text-to-Speech (TTS) engines and screen readers make it easier for people who have severe vision conditions to navigate sophisticated web pages and applications. Also, improving IT literacy has been the focus of initiatives dedicated to rendering people with disabilities independent in accessing and using technology. As a result, nowadays a large population of individuals who are blind can independently control most applications and websites on their smartphone or desktop device. Unfortunately, there still exists a large divide between the digital and physical world in terms of equitable access: unless they are considered public accommodations under the Americans with Disabilities Act or they are targeted to specific demographics, products are not required to demonstrate any accessibility features. As a result, their interface (e.g., buttons, knobs, display) is the result of a trade-off between form, functionality, and cost, which often makes it difficult to incorporate universal design principles. Moreover, despite several guidelines have been published and enforced (e.g., Title III), regulatory gaps leave space to non-compliance.

Although there are no general statistics about the level of universal access of products, single studies on specific types of devices revealed that most of them lack accessibility to individuals with disabilities. For instance, recent research on vending machines [2] demonstrated that it is impractical for people with sensory impairments to complete a purchase on specific types of eVending. The study analyzed product selection, payment, and collection in two common models of vending machines and found that poor User Interface design, lack of tactile information, and absence of non-visual feedback prevent the majority of customers with severe visual impairments from being able to independently select, pay, and collect a product. Similarly, most physical devices, from home appliances to technology gadgets, suffer from poor accessibility, especially to individuals who are blind. Consequently, while learning a device might require time and assistance, depending on the level of accessibility of its user interface, being able to independently use a product for the first time might be especially difficult, and, in some cases, impossible. Contributing factors include labels and messages that are not translated into tactile or auditory form (e.g., touch screens), components having counterintuitive affordances (e.g., credit card readers) or lacking touch or sound cues (e.g., capacitive keypads), missing support for error, and several other issues that also affect usability for individuals who are sighted. Several solutions have been attempted, and the promising work of ongoing research studies in multiple fields is producing new developments that have the potential to unlock access and simplify interaction. Nevertheless, the complexity of most system introduces further risks and challenges in terms of product feasibility, market penetration, and user adoption.

In this paper, we present the concept of Interface Digital Twin (IDT) and we discuss how it can be applied to create accessible replicas of the User Interface of a physical device that can be navigated with a smartphone using touch and audio. We describe the architecture of IDTs and we detail its integration in different types of existing technology using affordable resources that are already available, widespread, and easy to use.

II. Related Work

Although in the last decade developers have increasingly been realizing that accessibility is an important requirement of software and especially web pages, prior research [2] highlighted manufacturers' reluctancy to change the design of physical products, despite many lawsuits due to their non-compliance with guidelines and regulations in terms of equitable access, such as the American Disability Act. Unfortunately, this creates a divide between digital and physical technology that ultimately impacts the quality of life of people with disabilities and, specifically, individuals who are blind.

Consequently, several research groups in the field of assistive technology are addressing the lack of accessibility of physical devices: most of the work involves the development of new systems that incorporate haptic devices and leverage sensory substitution and multi-modal feedback to render visual information in tactile or audible form. Examples include smart canes combining computer vision or ultrasound systems and Machine Learning algorithms that provide users with information about the environment and help them in indoor and outdoor navigation tasks, hand-held systems reproduce shapes and text in tactile form, and wearable input and output devices that interact with the user via touch cues and vibration [3].

In the last decade, attention has been to Tangible Interfaces [4] [5] and wearable input and output systems [6] [7] [8], which refer to input and output devices that enable touch-based communication and interaction with applications on a computer, mobile device, or appliance. Indeed, they provide the benefit of translating visual information over the tactile channel, thus, making it potentially more accessible to individuals with vision impairments. Although tangible interfaces, such as the Tactile Shape Displays introduced in [9] can be utilized to represent simple shapes, more sophisticated devices are needed to support the complexity of the layout of a User Interface. Furthermore, as they require the development and production of new technology, they might not be sustainable given the numbers, market structure, and device regulations of assistive technology for people who are blind.

In [10], the authors propose the use of web-based interfaces to support people who are blind in interacting within a smart home environment. Although their solution incorporates accessible HTML tags and attributes that simplify control of basic home automation features, such as switching lights and checking their status, it might not be suitable for more sophisticated user interactions and for creating an accurate representation of the actual interface of a physical device or object.

A one-size-fits-all solution in the form of a Universal User Interface for individuals who are blind is presented by [11]: its architecture is based on a centralized database of ontologies and user models that works in combination with a dedicated mobile application. This, in turn, adapts interfaces depending on users' needs and displays them on the screen. Unfortunately, approaches that involve sophisticated architectures and complex implementation roadmaps, imply long time-to-market. Moreover, requiring end users to install additional software introduces risks in the deployment and adoption phases.

Solutions based on affordable systems that require very simple technology and little modifications to existing devices have been studied by several research groups. [12] suggests the use of digital fabrication tools to render the user interfaces of physical objects (e.g., home appliances) accessible thanks to tactile buttons and Braille labels printed in 3D. Moreover, the authors describe how leveraging crowdsourcing dynamics can enable individuals to create, share, and download models, for free, so that end users themselves can fix issues in home appliances and other devices.

Indeed, the accessibility challenge implies a trade-off between the urgency of the problem and the relevance of the solution, where the former favors low technical complexity to sophisticated technology and the latter is solely driven by the results in terms of user adoption.

III. System Design

The objective of our work is to develop an affordable and immediately actionable solution to the lack of accessibility of many products currently on the market. Specifically, our goal is to empower users and organizations focusing on accessibility to implement changes without requiring any intervention from the manufacturer. To this end, we applied a participatory design approach to the development of the system discussed in this paper [2]. Specifically, we involved groups of professionals, students, end users, and organizations devoted to disability, and we realized several workshops and studies to define the requirements and specifications of our solution. As an outcome of our preliminary work, users prioritized timely implementation, easy-to-learn technology, and low-cost interventions. Moreover, they urged fixes and patches that can be implemented by individual organizations and users, which they favored to more complete and powerful solutions that required sophisticated architectures, longer implementation roadmaps, and the involvement of multiple stakeholders.

As a result, the proposed solution consists of a web-based platform that enables users to design and share a digital replica of the user interface of a physical device, so that individuals who are blind can access it and use the accessibility features on their smartphones to navigate it, while receiving additional multi-modal guidance by the system. By doing this, they can build a mental model of the configuration of the physical interface of the actual device and learn how to operate it. For instance, while being in front of a vending machine, they can access the digital replica of its user interface with their mobile device: the text-to-speech features of their smartphone can provide them with a brief description of the main components; also, as they swipe their finger on the screen of their smartphone and browse the interface, tactile feedback and audio can be used to help them locate and read the label of buttons, receive instructions about how navigate the options of the payment system, simulate operations and obtain feedback that helps them learn how to make a selection, complete a purchase, and use the system, before or while they operate the physical device.

To this end, we introduce the concept of Interface Digital Twin, that is, a completely digital replica of a User Interface specifically designed to exactly mimic the form, layout, and workflow of a physical device, product, or object (e.g., a device or a paper document) to render it accessible and help users learn it. In contrast to Tangible Interfaces [4], IDTs aim at supporting multi-modal interaction via tactile icons and audible feedback without requiring a dedicated or external physical device. In addition to reproducing the original UI, the primary purpose of IDTs is to provide an additional information and interaction layer that improves usability and accessibility (e.g., by helping the user locate controls using vibration patterns and by translating visual labels in audible form) and adds user customization. As a result, its layout and features are aimed at supporting alternative multi-modal content that results in a more effective interaction with the associated object or device.

In general, IDTs are designed to be utilized when the user is in close proximity with the physical device or object they are associated with. Specifically, we expect individuals to primarily interact with them within in the Intimate (0 – 50cm), Personal (50cm – 1mt), or Social (1 – 4 mt) proxemic spaces. For instance, a user can download and access the IDT of the User Interface of a vending machine with their smartphone, by either scanning a tag or selecting the device among the ones available nearby (see Figure 1). To this end, IDTs should be compatible with different types of devices, screen sizes, and operating systems to account for the diverse requirements of diverse users. Simultaneously, standards must be defined to make them conveniently available for download and access, accurate in replicating the layout of the user interface of the physical device or object, and effective in making it accessible; also, they should be easy to design, publish, share, and reuse. Finally, they should enable affordable and versatile implementation with resource-effective systems that are accessible to everyone.

Therefore, the architecture of the proposed solution consists of three main components:

- a set of IDTs implemented based on standards and guidelines that ensure accessibility and functionality requirements are met;
- a system for conveniently retrieving and accessing the IDT of the physical device or object, especially in contexts of Just-In-Time interaction, such as purchasing from Vending Machines;
- a centralized repository facilitating the distribution of IDTs.

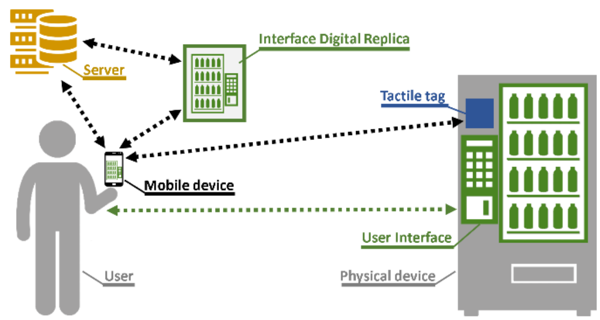

Figure 1. Example of interaction with an Interface Digital Replica: if a physical device having a User Interface that is not inherently accessible has an IDT associated to it, a tactile sign can be placed in a visible location (e.g., on its facade) to indicate the presence of the IDT. By doing this, when in proximity of the physical device, the user can sense and scan the IDT tag (e.g., QR code, NFC chip, or Bluetooth beacon). Consequently, the mobile device retrieves the Interface Digital Replica from the server and displays it on the screen of the smartphone, so that the user can navigate and understand the position and function of the components of the UI in the digital replica and operate them on the actual physical device.

A. *Interface Design and Implementation*

Several programming languages can be utilized to implement IDTs. However, based on feedback from our user-centered design processes, our system supports client-side web technologies, such as the HTML, Cascading Style Sheets (CSS), and JavaScript. Nowadays, thanks to HTML5 Application Programming Interfaces, browser integration, and external libraries and components, developers can implement the intangible version of an interface and make it interactive without requiring complex programming skills or sophisticated technical resources. By doing this, users are not required to install any specific application; also, using the web browser as a viewer increases compatibility, reduces implementation risks, provides a debugging environment, and helps troubleshooting potential issues. Modern languages enable the development of responsive web pages that dynamically adjust their content to fit any screen size, which, in turn, addresses device diversity. Existing elements, such as semantic tags and ARIA attributes, can conveniently be utilized to standardize implementation; also, currently available plug-ins for enabling touch-based interaction and gesture control can be used in combination with TTS and vibration features to provide individuals with multi-modal feedback. Specifically, these features can be utilized for reproducing some aspects of the affordances of the physical object by representing them into visual, audio, or tactile form. For instance, the UI of a vending machine can be replicated by a webpage in which HTML elements are positioned to match the configuration of the physical device (see Figure 2), so that when users move their fingers over the screen, they can sense the presence of buttons and listen to their labels in audio format. Also, in case of small devices or very complex UIs, the user can zoom in a specific part of the Interface.

In its simplest form, the IDT can replicate the layout of the physical UI of the device. In addition, IDTs can host further information about its components, such as how to operate them properly. For instance, previous studies found that individuals who are blind might find it difficult to collect their purchase from vending machines because the product release doors have anti-theft mechanisms that affect their accessibility in case vision is impaired. Similarly, in several devices, buttons support multiple actions depending on different ways to press them (e.g., short press versus long press). Therefore, the IDT could provide users with additional information about their operational details. Moreover, they can implement step-by-step wizards that offer guidance on how to complete the task, as if there was a human attendant assisting the user.

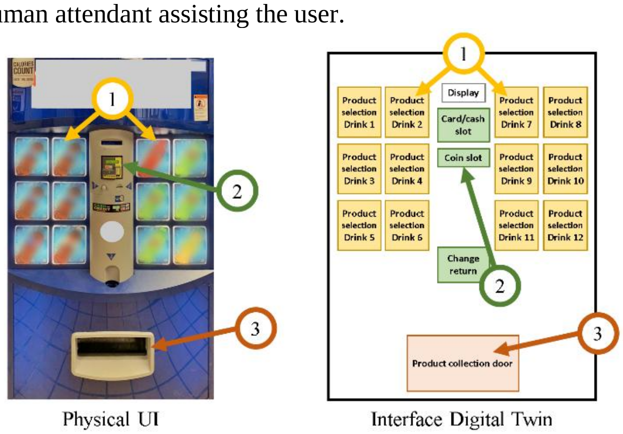

Figure 2. Example of the User Interface of a physical device, i.e., a vending machine (left), and its corresponding IDT (right). In addition to showing the location of the components of the UI, the IDT can organize them into categories that help the user understand their function, such as product selection (1), payment (2), and product collection (3). As users move their fingers on the display, the smartphone can vibrate or emit sounds to highlight the presence of an element, while its label can be displayed in audible form. Furthermore, the IDT can contain additional information on how to operate each component and can offer guidance on how to accomplish the task (i.e., complete a purchase).

In addition, custom UI building blocks and scripts can be designed to introduce more advanced features and make them uniformly available across multiple IDTs. Finally, as the languages utilized by the system inherently respond to Open source practices, developers can easily design new interfaces based on the components of existing IDTs. Specifically, in the case of devices, such as vending machines, in which models having the same physical structure and interface (invariant) are utilized to serve different products with diverse possible placement combinations (variant), the code of an IDT can be reused and modified to match the exact configuration and maintain consistency in terms of product selection.

Although for the purpose of this paper we mostly focus on accessibility aspects and on the client-side, IDTs can have a server-side component that enables additional features, such as retrieving more information about specific components or creating a Content Management System where users can dynamically update the content of the interface (e.g., the labels indicating product names on the UI of a vending machine). Also, IDTs act as separated containers: they do not share any information with one another. However, client-server interaction can be designed to enable IDTs to exchange information with the platform, such as the profile and characteristics of the user, which help customize the aspect and content of the IDT to their specific needs.

B. *Centralized Database of Interfaces and Guidelines*

The architecture of the system consists of a web platform that acts as a centralized repository that enables developers to create and share IDTs that users can download and access on their device.

Specifically, the platform has the purpose of defining the standards and describe the requirements, specifications, and implementation guidelines for the design of IDTs; by doing this, developers can produce IDTs that adhere to a common set of rules and protocols. In addition, the platform can host wikis, discussion boards, and other communication tools that support user-centered and participatory design practices: members who want to contribute to the project can discuss changes to the standards, promote updates and improvements, and share developers' guides and tutorials.

Secondly, the platform acts as a centralized hub for sharing and accessing IDTs. The codebase of IDTs can be hosted in separated repositories on external distributed Version Control Systems (e.g., GitHub, Bitbucket). This enables programmers to fork existing interfaces and reuse and modify their components, which is especially convenient in case of small changes, such as translating labels in multiple languages, or updating the configuration of buttons (e.g., diverse product configurations on different vending machines having the same model). There are several options for publishing the client-side of the IDT: it can be hosted on the developer's server, made it public on a code versioning system (e.g., GitHub pages), or uploaded on the platform itself. The actual webpage of the IDT can be made available via a dedicated URL on the platform. Moreover, each IDT can work as an independent sandbox, which increases security and provides developers with more flexibility. The developer will be responsible for hosting the server-side component of the IDT, if any. Regardless of where the client-side IDT is hosted, its link is associated with a redirect URL on the platform of the proposed system. The URL contains the handle of the developer, the brand of the manufacturer of the physical device or object, the product name, and its model, so that users can quickly search for an interface; in addition to specifying visibility settings, configuring accessibility criteria, and verifying that repositories meet the requirements, the platform keeps track of new versions and updates. The developer is responsible for any back-end scripts or databases that enable additional features with respect to the interactivity offered by client-side HTML, CSS, and JavaScript. This is to maximize interoperability and to avoid interfering with or over-regulating the many diverse systems utilized by different developers. Also, the platform can enable users to register an account, create their profile, and store their preferences. Also, the user can decide to share their preferences (e.g., presence and type of disability, preferred level of guidance, primary type of feedback) with IDTs, so that their components and interaction can be customized to better support their needs. Furthermore, the web platform can also store a history of devices the user interacted with, to facilitate access to frequently utilized IDTs.

Finally, the platform enables users and manufacturers to quickly associate IDTs with devices by supporting the creation of physical tags that can be placed on the physical object. In this regard, the platform can offer templates for tactile tags and automatically generate Quick Response (QR) codes or Near Field Communication (NFC) chips containing the URL of the IDT, to make it more convenient for users to access and control the interface, as discussed below. Also, it can provide developers with snippets and sample source code that help programmers quickly integrate IDTs in existing applications.

C. Convenient Access to and Control of User Interfaces

Each IDT can be associated with a unique URL that can be utilized to access their navigation. Moreover, the web address can be incorporated in a tag that enables users to easily retrieve the digital replica when they interact with the physical device. For instance, it can be in the form of a QR code or an NFC chip that can be positioned in a visible location of the physical product so that the tag can be conveniently scanned or read with a mobile phone. In order to render the tag easier to recognize to individuals who are blind, its area can be marked with a tactile pattern and it can be accompanied by a sign in Braille.

In addition, a wireless system (e.g., Bluetooth beacon) can be incorporated into the device so that users can discover the machines that are available nearby. This can be realized with dedicated mobile applications. Furthermore, physical products that are installed in a specific location, such as vending machines, can be associated with their geolocation coordinates or with their position in a building, floor or office so that people can use an app or a web browser to access a directory of the compatible devices nearby and visualize their interface via a navigation menu.

IV. Conclusions and Future Work

Despite existing accessibility guidelines require manufacturers to incorporate universal design features that enable individuals with different types of disabilities to use their products, regulatory gaps and the cost of updating existing technology make it very difficult to rapidly implement solutions that are applicable to systems currently on the market. As a result, people with disabilities are unable to independently use several devices, from basic home appliances to sophisticated eVending.

In this paper, we focused on improving the accessibility of physical devices, with specific regard to individuals who are blind. To this end, we presented a novel solution aimed at providing users with a system that can immediately render any interface easily accessible at a negligible cost, without requiring any intervention from the manufacturer. Specifically, we introduced the concept of IDTs, that is, replicas of the actual physical UI that the user can navigate thanks to the accessibility features of their smartphone. By doing this, they can understand the placement and function of the components of the interface, learn how to operate the device and accomplish their tasks. We detailed the architecture of the proposed system and we discussed how IDTs can be applied to any devices and products, thanks to its limited technical complexity.

In our future work, we will evaluate the effectiveness of the solution in rendering physical UIs accessible and supporting individuals with disabilities in independently using them.

References

[1] R. Varma, T.S. Vajaranant, B. Burkemper, S. Wu, M. Torres, C. Hsu, F. Choudhury, and R. McKean-Cowdin, "Visual impairment and blindness in adults in the United States: demographic and geographic variations from 2015 to 2050". JAMA ophthalmology, 134(7), pp.802-809. 2016.

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

[3] A. Bhowmick, and S.M. Hazarika, "An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends". Journal on Multimodal User Interfaces, 11(2), pp.149-172. 2017.

[4] M.S. Baldwin, R. Khurana, D. McIsaac, Y. Sun, T. Tran, X. Zhang, J. Fogarty, G.R. Hayes, and J. Mankoff. "Tangible Interfaces". In Web Accessibility (pp. 715-735). Springer, London. 2019.

[5] Caporusso, N., Mkrtchyan, L. and Badia, L., 2009. A multimodal interface device for online board games designed for sight-impaired people. IEEE Transactions on Information Technology in Biomedicine, 14(2), pp.248-254.

[6] Caporusso, N., 2008, May. A wearable Malossi alphabet interface for deafblind people. In Proceedings of the working conference on Advanced visual interfaces (pp. 445-448).

[7] 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.

[8] Caporusso, N., Biasi, L., Cinquepalmi, G., Trotta, G.F., Brunetti, A. and Bevilacqua, V., 2017, July. Enabling touch-based communication in wearable devices for people with sensory and multisensory impairments. In International Conference on Applied Human Factors and Ergonomics (pp. 149-159). Springer, Cham.

[9] A.F. Siu, "Advancing Accessible 3D Design for the Blind and Visually-Impaired via Tactile Shape Displays". In The Adjunct

Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology (pp. 146-149). 2019.

[10] M. Buzzi, B. Leporini, and C. Meattini, "Simple Smart Homes Web Interfaces for Blind People". In WEBIST (pp. 223-230). 2018.

[11] A. Khan, S. Khusro, and I. Alam, "Blindsense: An accessibility-inclusive universal user interface for blind people". Engineering, Technology & Applied Science Research, 8(2), pp.2775-2784. 2018.

[12] A. Guo, J. Kim, X.A. Chen, T. Yeh, S.E. Hudson, J. Mankoff, and J.P. Bigham, "Facade: Auto-generating tactile interfaces to appliances". In Proc of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 5826-5838)