

WearSkill: Personalized and Interchangeable Input with Wearables for Users with Motor Impairments

Ovidiu-Andrei Schipor

MintViz Lab, MANSiD Center
Ștefan cel Mare University of Suceava
Suceava, Romania
ovidiu.schipor@usm.ro

Laura-Bianca Bilius

MintViz Lab, MANSiD Center
Ștefan cel Mare University of Suceava
Suceava, Romania
laura.bilius@usm.ro

Radu-Daniel Vatavu

MintViz Lab, MANSiD Center
Ștefan cel Mare University of Suceava
Suceava, Romania
radu.vatavu@usm.ro

ABSTRACT

We introduce WearSkill, a web application that implements personalized and interchangeable input for wearable computing. We outline functional and quality requirements for WearSkill and present the engineering details of its implementation using web technology. We emphasize the interchangeability of input modalities with various wearables, e.g., touch input on a smart ring vs. mid-air gestures of the hand wearing a smartwatch vs. voice input detected by the microphone from a pair of smartglasses, towards personalized input for users with various motor abilities. Our findings, from a study involving twenty-one people with motor impairments, show that WearSkill can provide accurate recommendations for personalized input modalities that match 85.3% with users' own preferences.

CCS CONCEPTS

- **Human-centered computing** → **Accessibility technologies**;
- **Software and its engineering**;

KEYWORDS

Wearables, software architecture, event-based processing, touch input, gesture input, voice input, motor impairments, accessibility

ACM Reference Format:

Ovidiu-Andrei Schipor, Laura-Bianca Bilius, and Radu-Daniel Vatavu. 2022. WearSkill: Personalized and Interchangeable Input with Wearables for Users with Motor Impairments. In *19th Web for All Conference (W4A'22)*, April 25–26, 2022, Lyon, France. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3493612.3520455>

1 INTRODUCTION

Wearable devices have become increasingly available in the context of Ubicomp and IoT environments. With a global market of USD 40.65 billion in 2020 [3], wearables are becoming mainstream due to their many valuable functions, e.g., health and fitness tracking, and integration with other mainstream devices, such as smartphones. As end users who are part of this market, people with motor impairments experience challenges interacting with wearables [22] that were not designed to be accessible in the first place. The type

and severity of the motor condition determine the specificity of such challenges [9,13,16,31] with impact on the feasibility of the input modalities that can be effectively used to interact with off-the-shelf devices, such as touch input [26], hand gestures [8], foot gestures [6], or movements of the head [24]. For example, Mott *et al.* [13] reported barriers regarding the physical accessibility of VR devices, such as putting HMDs on and taking them off, adjusting the head strap, or maintaining view of the controllers, and Malu *et al.* [9] reported challenges regarding button, swipe, and tap-based interactions for smartwatch input. A workaround is system design that features alternative input modalities that best match individual users' abilities [33]. Personalization and interchangeability of input modalities are also useful for interactions performed in conditions characterized by situational impairments [1,5,20].

We are interested in this work in software architecture design for wearables featuring personalization and interchangeability of input modalities. Our contributions are as follows: (1) a set of software design requirements for input personalization and interchangeability with wearables; (2) WearSkill, our web-based application that implements our set of requirements towards personalized, multi-device, multi-modality input with wearables; and (3) a study with N=21 people with motor impairments showing 85.3% accuracy for predicting users' preferences for wearable devices and input modalities that best match their motor abilities.

2 RELATED WORK

Prior work has documented challenges experienced by people with motor impairments during interactions with computer systems of many kinds. From mouse [32], keyboard [4], and remote control [25] input to interactions with mobile devices [12,27,30] to large touchscreen displays [14] to wearables [9,10,31], a wide range of devices and input modalities have been scrutinized for accessibility. Among these, accessible interactions with wearables have been addressed to a lesser extent: in their systematic literature review of wearable interactions for users with motor impairments, Șean and Vatavu [22] reported that hand gestures have been disproportionately favored (41.6%) compared to other input modalities, e.g., head gestures (23.4%) or voice input (13.0%). Based on their findings, Șean and Vatavu proposed the WISE framework, a set of recommendations to increase the accessibility of wearables: (W) exploring diverse designs of wearables, (I) new input modalities and techniques for accessible wearable interactions, (S) more user studies and evaluations, and (E) extending wearable interactions with other devices. Due to its direct connection to our scope, we adopt the WISE framework as the conceptual foundation for WearSkill.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

W4A'22, April 25–26, 2022, Lyon, France

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9170-2/22/04...\$15.00

<https://doi.org/10.1145/3493612.3520455>

Several unique characteristics of wearables make them suitable for personalized and interchangeable input: diverse form factors, placement close to or in contact with the body, and varied sensing options to detect user input, including taps and touch input, movements of the fingers and hands, voice, eye gaze, and others; see [21,22]. Also, multiple devices can be worn together, e.g., a smartwatch and a smart ring, enabling the use of one or another according to context. This flexibility, however, raises several technical challenges for software architecture designs and for the algorithms employed to recognize user action across multiple devices.

Various technical solutions have been proposed for input with heterogeneous devices [18,19]. One example is Euphoria, a scalable, event-driven software architecture for implementing interactions across heterogeneous devices in smart environments. Euphoria consists of five software layers: Producers, Emitters, Engine, Receivers, and Consumers. The outermost components, represented by Producers and Consumers, implement the specific details required by different platforms, operating systems, and APIs. In the center of the architecture, Emitters and Receivers abstract the strategy for exchanging messages with all devices sharing the same interface and communicating via the same protocol. The business logic for the communications resides at the Engine level, where messages are routed between devices. In this work, we reuse the open-source Euphoria [18] as the middleware for implementing WearSkill.

3 WEARSKILL

We introduce WearSkill, our web-based application that implements personalized and interchangeable input with wearables. We start by presenting functional requirements for WearSkill in accordance with the WISE framework [22]. We then leverage the SQuaRE [7] model to select six quality requirements, which we implement with three technological and three design patterns approaches.

3.1 Design Requirements

Based on the directions set by the WISE framework [22], we formulate four functional requirements (F_1 to F_4) for WearSkill:

- F_1 . *Wearables*. WearSkill must be flexible to integrate a variety of wearables and their integration must require only minimal software changes at the outer layer of the software architecture. We consider wearable devices that have built-in Wi-Fi connection and support communication protocols for the web, such as WebSocket.¹
- F_2 . *Interactions*. WearSkill facilitates execution of system functions on various output devices, e.g., a PC or a TV set, with input performed with wearables. To this end, users perform a command with the wearable they prefer in a given context. WearSkill stores associations between commands performed with wearables and system functions of output devices, e.g., a wave of the hand wearing a smartwatch turns on the TV.
- F_3 . *Studies*. WearSkill offers out-of-the-box support for logging input data to enable more studies to learn about how users employ wearables and how they personalize input.
- F_4 . *Extension*. The main goal of WearSkill is to enable personalized and interchangeable input with wearables as the middleware for a distributed user interface [28].

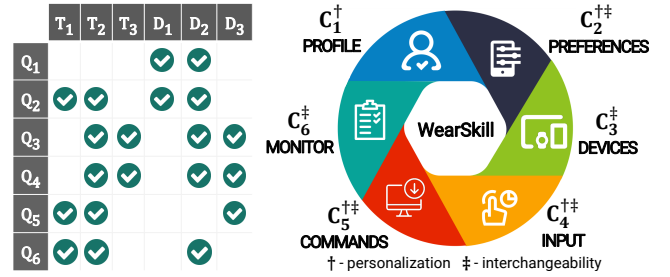


Figure 1: Design requirements for WEARSKILL (left) and an overview of its software components (right) for personalization and interchangeability of input with wearables.

We choose six quality requirements from the ISO/IEC 25010:2011 SQuaRE [7] model (Systems and Software Quality Requirements and Evaluation) that are directly relevant to wearables [17] and to the scope of the WearSkill application, respectively:

- Q_1 . *Modularity* refers to splitting the software system into discrete units that interact with each other through interfaces. This requirement acknowledges the heterogeneity of wearables (F_1) and keeps the adapter layer as thin as possible.
- Q_2 . *Reusability* indicates the degree to which a specific asset can be reused in another configuration of the system. Since WearSkill enables input with heterogeneous devices (F_1), developers that build on WearSkill can reuse existing functionality to accommodate new wearables and output devices.
- Q_3 . *Interoperability* is the ability of two or more modules to communicate with each other based on a common standard. WearSkill integrates different types of wearables (F_1) by means of dedicated modules that employ standard communication protocols to enable a variety of interactions (F_2).
- Q_4 . *Replaceability* refers to the possibility to replace a component with another, equivalent one. The interchangeability of wearables enables the use of different input devices (F_1 , F_4) to execute a given system function on an output device.
- Q_5 . *Appropriateness* indicates the degree to which the software system meets end-user needs in a personalized way. This property connects to the functional requirements F_3 and F_4 .
- Q_6 . *Learnability* specifies the capacity of the system to help the user transition from novice to expert mode. Also, better understanding end-user needs for wearable interactions (F_3) can lead to better future versions of WearSkill (F_4).

The first two quality requirements, Q_1 and Q_2 , set the condition for low-level software modules, requirements Q_3 and Q_4 refer to high-level software components, and Q_5 and Q_6 specify the characteristics of the interaction between users and WearSkill. To implement these quality requirements, we adopt three technological (T_1 to T_3) and three design patterns (D_1 to D_3) approaches; see Figure 1, left. Due to the high portability of web applications, we choose web technology (T_1) for the development of the WearSkill user interface. We also choose JavaScript as our programming language of choice (T_2) since it runs on a variety of platforms, either in a web browser or a runtime environment. Communications between the software components are implemented with Euphoria [18] and, thus, employ

¹https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

HTTP and WebSocket protocols (T_3). In terms of design patterns, we establish WearSkill on the SOLID principles [11]. We use dependency injection, *i.e.*, injecting of dependencies rather than creating them inside the object, to obtain a strong separation between object creation and utilization (design pattern D_1). Also, WearSkill is divided into several layers (models, views, view models, presenters, controllers, repositories, and gateways) that communicate through interfaces (D_2). To simulate usage scenarios (D_3), we employ high coverage end-to-end testing. To implement the functional requirements F_1 to F_4 , we designed six software components for WearSkill (C_1 to C_6 in Figure 1), presented in the next subsection.

3.2 Implementation

The software components of WearSkill play specific roles in achieving input personalization and interchangeability. Figure 1, right illustrates the interaction flow, from the specification of the user profile to runtime monitoring of the system, as follows:

- C_1 . PROFILE enables users to enter personal data (in accordance with the functional requirement F_2) represented by their motor symptoms, *e.g.*, slow movements, low strength, etc., which we adopted from [2]; see Figure 2, left.
- C_2 . PREFERENCES employs recommendations generated by a machine learning model that uses information from PROFILE to suggest wearables and input modalities (according to functional requirements F_1 and F_2). In the current version, we implemented support for three types of wearables (smartwatches, smartglasses, and smart rings) and four input modalities (touch input, hand motion, head motion, and voice input), but extensions can be further integrated.
- C_3 . Registration of input and output devices is performed under the DEVICES section (requirements F_1 and F_4). Each device is identified by its MAC address and associated with one of the following categories: smartwatch, smartglasses, smart ring, or a generic output device, on which system functions will execute following input with the first three types of devices.
- C_4 . The INPUT component is in charge of detecting and recognizing touch, motion, and voice input (functional requirements F_1 and F_2) that users can particularize by providing training samples; see Figure 2, right. Voice input consists in word commands recognized using the Google Speech-to-Text API. Touch stroke-gesture and motion input are implemented with the $\$P$ point-cloud gesture recognizer [29].
- C_5 . Input and output devices are associated in the COMMANDS section (requirement F_4). Commands run on output devices, *e.g.*, turn on the TV. The same command can be entered with different input modalities to implement interchangeability.
- C_6 . Users can supervise the entire process from the RUNTIME MONITOR component (functional requirement F_3). For example, if input from a given device is not recognized or system functions are not executed on the output device, warnings help the user in identifying the cause.

We developed the user interface on top of Vue.js,² a progressive JavaScript framework that enables clear separation between views and view-models. Back-end models run on the Node.js³ platform

and communicate with the front-end and the input and output devices via HTTP and WebSocket protocols. For end-to-end testing, we employed Cypress,⁴ and specified more than 200 tests covering usual scenarios. To test WearSkill, we developed client web applications for three wearables: Samsung Galaxy Watch 3, Gear Fit 2 (which we mounted on a custom 3D printed support to be used as a smart ring), and the Vuzix Blade smartglasses. Next, we focus on the recommender implemented under the PREFERENCES component to match wearables and input modalities to users' motor abilities.

3.3 Personalized Recommendations for Input Modalities with WearSkill

The PREFERENCES software component integrates a machine learning model that employs eleven self-reported motor symptoms, entered by the user as yes/no responses when setting their WearSkill profile (Figure 2, left), to suggest recommendations for wearables and input modalities that best match users' motor abilities. The symptoms are: slow movements, spasm, low strength, tremor, poor coordination, rapid fatigue, difficulty gripping, difficulty holding, lack of sensation, difficulty controlling direction, and difficulty controlling distance, which we adopted from [2]. We conducted a user study to collect the data needed to train this software component.

3.3.1 Participants. A number of $N=21$ people with motor impairments (spinal cord injury located at various vertebrae, spina bifida, traumatic brain injury), aged between 28 and 59 years old ($M=43.3$, $SD=8.2$ years), were recruited via a non-profit organization providing technical assistance to people with disabilities.

3.3.2 Method. We used an online questionnaire to elicit preference ratings for various combinations of wearables and input modalities using 5-point Likert scales with items ranging from 1 (not suitable for me) to 5 (very suitable). Our experiment was a within-subjects design with two independent variables: input modality (nominal variable with four conditions—touch input, hand motion, head motion, and voice input) and wearable (nominal variable with three conditions—smartwatch, smartglasses, and smart ring).

3.3.3 Dataset. We collected a total number of 21 (participants) $\times$ 11 (symptoms) = 231 records representing *predictors* and 21 (participants) $\times$ 3 (wearables) $\times$ 4 (input modalities) = 252 preferences representing *recommendations* for the PREFERENCES component.

3.3.4 Results. We employed Scikit-learn [15] to evaluate various classifiers on our dataset using hyperparameter tuning with grid search optimization [23] and leave-one-out cross-validation. Our choice of algorithms included linear models and discriminant analysis, Nearest Neighbor classifiers, decision trees, ensemble methods, semi-supervised models, neural networks, and Support Vector Machines models. During training, we grouped participants' ratings obtained with the 5-point Likert-scales in two classes: *do not recommend* (corresponding to the Likert scale items 1 and 2) and *recommend* (items 3, 4, and 5), respectively. Hyperparameter tuning enabled us to cover a wide range of configurations for the classifiers consider in our evaluation, such as regarding the number of neighbors to use for the K Nearest Neighbors classifier or the best solver

²<https://vuejs.org>

³<https://nodejs.org/en>

⁴<https://www.cypress.io>

Table 1: Accuracy rates for predicting users' preferences for wearables and input modalities from their motor symptoms.

Classifier	Smartwatch				Smartglasses				Smart ring				Mean accuracy
	Touch input	Hand motion	Head motion	Voice input	Touch input	Hand motion	Head motion	Voice input	Touch input	Hand motion	Head motion	Voice input	
Decision Tree [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	57.1%	90.5%	95.2%	100%	95.2%	100%	57.1%	85.3%
Gradient Boosting [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	57.1%	90.5%	95.2%	100%	95.2%	100%	57.1%	85.3%
Ada Boost [†] [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	57.1%	90.5%	95.2%	100%	95.2%	100%	57.1%	85.3%
KNeighbors [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	52.4%	90.5%	95.2%	100%	95.2%	100%	57.1%	84.9%
Label Propagation [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	52.4%	90.5%	95.2%	100%	95.2%	100%	57.1%	84.9%
Radius Neighbors [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	52.4%	90.5%	95.2%	100%	95.2%	100%	52.4%	84.5%
Logistic Regression [ref.]	85.7%	90.5%	76.2%	81.0%	90.5%	47.6%	90.5%	95.2%	100%	95.2%	100%	61.9%	84.5%
Multinomial Naïve Bayes [ref.]	85.7%	90.5%	81.0%	81.0%	90.5%	52.4%	90.5%	95.2%	100%	95.2%	100%	47.6%	84.1%
Label Spreading [ref.]	71.4%	90.5%	81.0%	81.0%	90.5%	57.1%	90.5%	95.2%	100%	95.2%	100%	47.6%	83.3%

[†]The base estimator was a Random Forest [ref.] classifier.

The figure consists of two side-by-side screenshots of the WearSkill web application. The left screenshot shows the 'Profile' tab, which includes a user profile section with fields for Name, Email, Birth year, and Gender. Below this is a section titled 'Which of the following situations apply to you?' with a list of motor symptoms and checkboxes. The right screenshot shows the 'Touch input' tab, which includes a section for defining a touch gesture (e.g., 'swipe-right') and a list of training examples (e.g., 'double-tap', 'swipe-left', 'swipe-right', 'tap'). It also includes a section for adding new training instances with dropdowns for 'Select motion input' and 'Select device to perform touch input'.

Figure 2: Snapshots of the WearSkill application. On the left, symptoms of the motor condition enable generation of personalized recommendations of input modalities for wearables. On the right, a personalized gesture is provided for touch input.

for the logistic regression. An example of applying this technique, expressed with the formalism of Skikit-learn [15], is:

```
GradientBoostingClassifier (subsample = 1, validation_fraction = 0.7, random_state = 0, criterion = 'mae', n_estimators = 19, min_samples_split = 2, min_samples_leaf = 11, max_leaf_nodes = 3, max_features = 'auto', max_depth = 3, loss = 'exponential', learning_rate = 1.84).
```

Table 1 presents the results obtained with leave-one-out cross-validation, which show good performance achieved by all of the classifiers evaluated in our study. The maximum accuracy for predicting users' preferences regarding personalized input modalities with wearables was 85.3% when averaged across all of the wearable and input modality conditions (last column from Table 1).

4 CONCLUSION AND FUTURE WORK

We presented WearSkill, a web-based application that implements input personalization and interchangeability for wearables. In this paper, we focused on the engineering details of WearSkill, but future work is envisaged to evaluate its usability. We also plan to continue the development of WearSkill (<http://www.eed.usv.ro/mintviz/projects/WearSkill>) towards an open-source solution on the web that integrates with IoT devices.

ACKNOWLEDGMENTS

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, project no. PN-III-P2-2.1-PED-2019-0352 (276PED/2020), within PNCDI III.

REFERENCES

- [1] Elgin Akpınar, Yeliz Yeşilada, and Selim Temizer. 2020. The Effect of Context on Small Screen and Wearable Device Users' Performance - A Systematic Review. *ACM Comput. Surv.* 53, 3, Article 52 (may 2020), 44 pages. <https://doi.org/10.1145/3386370>
- [2] Leah Findlater, Alex Jansen, Kristen Shinohara, Morgan Dixon, Peter Kamb, Joshua Rakita, and Jacob O. Wobbrock. 2010. Enhanced Area Cursors: Reducing Fine Pointing Demands for People with Motor Impairments. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 153–162. <https://doi.org/10.1145/1866029.1866055>
- [3] Grand View Research. 2021. *Wearable Technology Market Size, Share & Trends Analysis Report By Product (Wrist-Wear, Eye-Wear & Head-Wear, Foot-Wear, Neck-Wear, Body-wear), By Application, By Region, And Segment Forecasts, 2021-2028*. Grand View Research. <https://www.grandviewresearch.com/industry-analysis/wearable-technology-market>
- [4] Yizheng Gu, Chun Yu, and Yuanchun Shi. 2019. The Dynamic Grouping Keyboard: A General Keyboard Optimization Approach for Users with Motor Impairment. *CCF Transactions on Pervasive Computing and Interaction* 1, 2 (2019), 89–99. <https://doi.org/10.1007/S42486-019-00010-5>
- [5] Florian Heller, Davy Vanacken, Eva Geurts, and Kris Luyten. 2020. Impact of Situational Impairment on Interaction with Wearable Displays. In *Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '20)*. ACM, New York, NY, USA, Article 21, 5 pages. <https://doi.org/10.1145/3406324.3410540>
- [6] Xiaozhu Hu, Jiting Wang, Weiwei Gao, Chun Yu, and Yuanchun Shi. 2021. FootUI: Assisting People with Upper Body Motor Impairments to Use Smartphones with Foot Gestures on the Bed. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 436, 7 pages. <https://doi.org/10.1145/3411763.3451782>
- [7] International Organization for Standardization. 2011. *ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuARE) - System and software quality models*. International Organization for Standardization. <https://www.iso.org/standard/35733.html>
- [8] Yiqin Lu, Bingjian Huang, Chun Yu, Guahong Liu, and Yuanchun Shi. 2020. Designing and Evaluating Hand-to-Hand Gestures with Dual Commodity Wrist-Worn Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1, Article 20 (2020), 27 pages. <https://doi.org/10.1145/3380984>
- [9] Meethu Malu, Pramod Chundury, and Leah Findlater. 2018. Exploring Accessible Smartwatch Interactions for People with Upper Body Motor Impairments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3174062>
- [10] Meethu Malu and Leah Findlater. 2015. Personalized, Wearable Control of a Head-Mounted Display for Users with Upper Body Motor Impairments. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 221–230. <https://doi.org/10.1145/2702123.2702188>
- [11] Robert C. Martin. 2018. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, Boston, MA, USA. <https://www.pearson.com/us/higher-education/program/Martin-Clean-Architecture-A-Craftsman-s-Guide-to-Software-Structure-and-Design/PGM333762.html?tab=authors>
- [12] Kyle Montague, Hugo Nicolau, and Vicki L. Hanson. 2014. Motor-Impaired Touchscreen Interactions in the Wild. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '14)*. ACM, New York, NY, USA, 123–130. <https://doi.org/10.1145/2661334.2661362>
- [13] Martez Mott, John Tang, Shaun Kane, Edward Cutrell, and Meredith Ringel Morris. 2020. "I Just Went into It Assuming That I Wouldn't Be Able to Have the Full Experience": Understanding the Accessibility of Virtual Reality for People with Limited Mobility. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20)*. ACM, New York, NY, USA, Article 43, 13 pages. <https://doi.org/10.1145/3373625.3416998>
- [14] Martez E. Mott, Radu-Daniel Vatavu, Shaun K. Kane, and Jacob O. Wobbrock. 2016. Smart Touch: Improving Touch Accuracy for People with Motor Impairments with Template Matching. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1934–1946. <https://doi.org/10.1145/2858036.2858390>
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. <https://arxiv.org/abs/1201.0490>
- [16] Maria Doina Schipor. 2019. Attitude and Self-Efficacy of Students with Motor Impairments Regarding Touch Input Technology. *Revista Românească pt Educație Multidimensională* 11, 1 (2019), 177–186. <http://dx.doi.org/10.18662/trem/104>
- [17] Ovidiu-Andrei Schipor and Radu-Daniel Vatavu. 2021. Software Architecture Based on Web Standards for Gesture Input with Smartwatches and Smartglasses. In *Proceedings of the 20th International Conference on Mobile and Ubiquitous Multimedia (MUM '21)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3490632.3497780>
- [18] Ovidiu-Andrei Schipor, Radu-Daniel Vatavu, and Jean Vanderdonckt. 2019. Euphoria: A Scalable, Event-driven Architecture for Designing Interactions Across Heterogeneous Devices in Smart Environments. *Information and Software Technology* 109 (2019), 43–59. <https://doi.org/10.1016/j.infsof.2019.01.006>
- [19] Ovidiu-Andrei Schipor, Radu-Daniel Vatavu, and Wenjun Wu. 2019. SAPIENS: Towards Software Architecture to Support Peripheral Interaction in Smart Environments. *Proc. ACM Hum.-Comput. Interact.* 3, EICS, Article 11 (jun 2019), 24 pages. <https://doi.org/10.1145/3331153>
- [20] Andrew Sears, Min Lin, Julie Jacko, and Yan Xiao. 2003. When Computers Fade... Pervasive Computing and Situationally-Induced Impairments and Disabilities. In *Proceedings of the 10th Int. Conference on Human-Computer Interaction (HCI International '03)*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1298–1302.
- [21] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanachana Thilakarathna, Mahbub Hassan, and Aruna Seneviratne. 2017. A Survey of Wearable Devices and Challenges. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2573–2620. <https://doi.org/10.1109/COMST.2017.2731979>
- [22] Alexandru-Ionut Sean and Radu-Daniel Vatavu. 2021. Wearable Interactions for Users with Motor Impairments: Systematic Review, Inventory, and Research Implications. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York, NY, USA, Article 7, 15 pages. <https://doi.org/10.1145/3441852.3471212>
- [23] Huy Tu and Vivek Nair. 2018. Is One Hyperparameter Optimizer Enough?. In *Proceedings of the 4th ACM SIGSOFT International Workshop on Software Analytics*. ACM, New York, NY, USA, 19–25. <https://doi.org/10.1145/3278142.3278145>
- [24] Ovidiu-Ciprian Ungurean, Stefan-Gheorghe Pentiu, and Radu-Daniel Vatavu. 2009. Use Your Head: An Interface for Computer Games using Head Gestures. In *Proceedings of the 8th International Gesture Workshop (GW '09)*. Bielefeld University, Bielefeld, Germany, 2 pages.
- [25] Ovidiu-Ciprian Ungurean and Radu-Daniel Vatavu. 2021. Coping, Hacking, and DIY: Reframing the Accessibility of Interactions with Television for People with Motor Impairments. In *Proceedings of the ACM International Conference on Interactive Media Experiences (IMX '21)*. ACM, New York, NY, USA, 37–49. <https://doi.org/10.1145/3452918.3458802>
- [26] Ovidiu-Ciprian Ungurean and Radu-Daniel Vatavu. 2021. Users with Motor Impairments' Preferences for Smart Wearables to Access and Interact with Ambient Intelligence Applications and Services. In *Proceedings of the 12th International Symposium on Ambient Intelligence (ISAmI '21)*. Springer, Switzerland, 10 pages.
- [27] Ovidiu-Ciprian Ungurean, Radu-Daniel Vatavu, Luis A. Leiva, and Réjean Plamondon. 2018. Gesture Input for Users with Motor Impairments on Touchscreens: Empirical Results Based on the Kinematic Theory. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, 1–6. <https://doi.org/10.1145/3170427.3188619>
- [28] Jean Vanderdonckt. 2010. Distributed User Interfaces: How to Distribute User Interface Elements Across Users, Platforms, and Environments. *Proceedings of the XIth Congreso Internacional de Interacción Persona-Ordenador 20 (2010)*, 3–14. <http://www.usixml.org/en/vanderdonckt-j-distributed-user-interfaces-how-to-distribute-user-interface-elements-across-users-platforms-and-environm.html?IDC=465&IDD=1433>
- [29] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures as Point Clouds: A SP Recognizer for User Interface Prototypes. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI '12)*. ACM, New York, NY, USA, 273–280. <https://doi.org/10.1145/2388676.2388732>
- [30] Radu-Daniel Vatavu and Ovidiu-Ciprian Ungurean. 2019. Stroke-Gesture Input for People with Motor Impairments: Empirical Results & Research Roadmap. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–14. <https://doi.org/10.1145/3290605.3300445>
- [31] Radu-Daniel Vatavu and Ovidiu-Ciprian Ungurean. 2022. Understanding Gesture Input Articulation with Upper-Body Wearables for Users with Upper-Body Motor Impairments. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '22)*. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3491102.3501964>
- [32] Jacob O. Wobbrock and Krzysztof Z. Gajos. 2008. Goal Crossing with Mice and Trackballs for People with Motor Impairments: Performance, Submovements, and Design Directions. *ACM Trans. Access. Comput.* 1, 1, Article 4 (may 2008), 37 pages. <https://doi.org/10.1145/1361203.1361207>
- [33] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. *ACM Trans. Access. Comput.* 3, 3, Article 9 (apr 2011), 27 pages. <https://doi.org/10.1145/1952383.1952384>