



**HAL**  
open science

# Une approche de programmation par démonstration pour l'intégration de la multimodalité sous mobile

Zouhir Bellal, Nadia Elouali, Sidi Mohamed Benslimane

► **To cite this version:**

Zouhir Bellal, Nadia Elouali, Sidi Mohamed Benslimane. Une approche de programmation par démonstration pour l'intégration de la multimodalité sous mobile. 29ème conférence francophone sur l'Interaction Homme-Machine, AFIHM, Aug 2017, Poitiers, France. 10 p., 10.1145/3132129.3132154 . hal-01578219

**HAL Id: hal-01578219**

**<https://hal.science/hal-01578219>**

Submitted on 28 Aug 2017

**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.

---

# Une approche de programmation par démonstration pour l'intégration de la multimodalité sous mobile

**Zouhir Bellal**

École Supérieure en Informatique, Sidi  
Bel-Abbès, Laboratoire LABRI  
Sidi Bel-Abbès 22000, Algérie  
z.bellal@esi-sba.dz

**Nadia Elouali**

École Supérieure en Informatique, Sidi  
Bel-Abbès, Laboratoire LABRI  
Sidi Bel-Abbès 22000, Algérie  
n.elouali@esi-sba.dz

**Sidi Mohamed Benslimane**

École Supérieure en Informatique, Sidi  
Bel-Abbès, Laboratoire LABRI  
Sidi Bel-Abbès 22000, Algérie  
s.benslimane@esi-sba.dz

**ABSTRACT**

The emergence of sensor-based modalities enriches the human-mobile interaction. However, mobile applications end-users are often faced with interaction restrictions and do not yet have the possibility to modify their interaction modalities, according to their needs. In this paper, we present an approach that allows smartphone users to add and modify the interaction modalities of their already installed mobile applications, according to their preferences and needs. The approach is based on a Wrapper

---

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the Owner/Author.

IHM'17, August 29-September 1, 2017, Poitiers, France

© 2017 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-5109-6/17/08. <https://doi.org/10.1145/3132129.3132154>

that acts as an intermediary between the user and the application and on the Programming By Demonstration (PBD) that allows the user to specify the desired changes.

#### CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Mobile computing**; *Graphical user interfaces*; *User interface programming*;

#### KEYWORDS

Programming By Demonstration(PBD, End-user development, MultiModal User Interfaces, Mobile Applications, Software Modernization

#### RÉSUMÉ

Ces dernières années, la présence de nouveaux capteurs sur les appareils mobiles a suscité un grand intérêt. L'émergence des modalités basées sur ces capteurs enrichit l'interaction homme-mobile. Cependant, les utilisateurs finaux des applications mobiles rencontrent souvent des restrictions d'interaction (petits écrans, clavier virtuel inconfortable, l'obligation d'utiliser les yeux et les deux mains pour certaines interactions, etc) et n'ont pas encore la possibilité de modifier leurs modalités d'interaction, selon leurs besoins. Dans ce papier, nous présentons une approche qui permet aux utilisateurs des smartphones, qui ne sont pas forcément des développeurs, d'ajouter et modifier, selon leurs préférences et besoins, les modalités d'interaction des applications déjà installées sur leurs smartphones. L'approche se base sur, d'une part, l'utilisation d'un Wrapper qui joue le rôle d'intermédiaire entre l'utilisateur et l'application et sur, d'autre part, la programmation par démonstration qui permet à l'utilisateur de préciser les modifications souhaitées.

#### MOTS-CLEFS

Programmation par démonstration, Interfaces MultiModales, Applications Mobiles, Modernisation des logiciels

#### ACM Reference Format :

Z. Bellal, N. Elouali and S.M. Benslimane 2017. Une approche de programmation par démonstration pour l'intégration de la multimodalité sous mobile. In *Proceedings of 29<sup>ème</sup> Conférence Francophone sur l'Interaction Homme-Machine, Poitiers Futuroscope, FR Août 2017 (IHM 2017)*, 10 pages.

DOI : 10.1145/3132129.3132154

## 1 INTRODUCTION

Les interfaces multimodales sous mobile cherchent à exploiter à la fois les capacités humaines de communication (la parole, le geste, le toucher, l'expression du visage, etc.) et la puissance des smartphones qui a donné naissance à de nouveaux types d'interaction (vocale, tactile, orientation et secouage du téléphone, etc.).

De nombreux outils dans la littérature cherchent à faciliter le développement des applications multimodales [16][13][17][22]. Cependant, aucun outil ne permet d'intégrer les interactions basées sur les capteurs mobiles ainsi que leurs combinaisons, selon les besoins de l'utilisateur.

Cela représente un obstacle devant la propagation de la multimodalité sous mobile et ne lui donne pas l'opportunité d'être proche de ses utilisateurs finaux.

Dans cet article, nous présentons l'architecture de notre nouvelle approche (Fig. 1). C'est une approche qui permet aux utilisateurs des smartphones, qui ne sont pas forcément des utilisateurs avertis, d'ajouter et/ou modifier, selon leurs préférences/besoins, les modalités d'interaction des applications déjà installées sur leurs smartphones. Elle se base sur l'utilisation d'un Wrapper qui joue le rôle d'intermédiaire entre l'utilisateur et l'application et sur la programmation par démonstration qui permet à l'utilisateur de préciser la modification/ l'ajout souhaité.

## 2 ÉTAT DE L'ART

Nous avons classifié les travaux existants en trois catégories :

- La première catégorie regroupe les outils qui facilitent le développement des applications mobiles multimodales.
- La seconde regroupe les approches qui donnent la possibilité d'intégrer des nouvelles modalités dans les systèmes existants.
- La dernière présente les différents travaux qui intègrent l'utilisateur final (non expert) dans le processus de développement de l'interaction.

La première catégorie présente un nombre important de travaux :

ACICARE [27] est une plateforme qui permet la création et l'évaluation des interfaces multimodales sur les téléphones mobiles (mais pas de type smartphone ou tablette [1]). Cette plateforme combine deux autres plateformes : ICARE [5] qui représente une plateforme à base de composant pour le développement rapide des interfaces multimodales et ACIDU [10] qui permet de capturer les données d'usage sur téléphones. Ces plateformes sont limitées en termes de modalités supportées, car ils ne supportent que la reconnaissance vocale et l'interaction avec le clavier [26].

MIMIC (Mobile Multimodality Creator)[14] est un Framework à base de modèles pour la construction d'applications mobiles multimodales. Il permet la modélisation graphique des interactions multimodales en entrée et en sortie avec un langage de modélisation appelé M4L (Mobile MultiModality

Modeling Language) ainsi que la génération automatique du code de ces interactions (sous Android, iOS et HTML5/CSS3)[13]. Le Framework supporte plusieurs modalités en entrée (tactile, vocale, accélération, orientation, etc.) et en sortie (affichage, synthèse vocale, vibration) ainsi que leurs combinaisons, selon les propriétés CARE (Complémentarité, Assignment, Redondance et Équivalence)[8]. Le même objectif est visé par le Framework M3I [24]. Il se base sur une bibliothèque Java (pour Android) et une approche à base des règles pour simplifier la création des applications mobiles multimodales adaptées au contexte. M3I ne supporte que les modalités d'interaction en entrée (gestuelle, tactile, vocal)[23].

Les outils présentés dans cette catégorie sont destinés aux développeurs et ne donnent pas la possibilité de modifier les modalités des applications existantes. La modification des modalités dans des applications déjà générées sous MIMIC, par exemple, nécessite une re-modélisation de l'application et engendre l'écrasement du code de la version initiale [12].

Contrairement à la première catégorie, la deuxième présente les approches qui permettent l'intégration des modalités dans des systèmes déjà existants.

Screen scraping [6][7] est l'approche la plus connue dans la modernisation des interfaces. Elle permet d'encapsuler les anciens systèmes à base interfaces de ligne de commande et les rendre accessible à partir d'une interface graphique ou web. Elle utilise un Wrapper qui permet de mapper entre l'ancienne et la nouvelle interface. La même méthode d'encapsulation a été utilisée dans l'approche Application wrapper [20][19]. Elle permet d'intégrer les interactions gestuelles et vocales, capturées à l'aide d'un Kinect, dans les applications desktop. Les interactions natives de l'application sont extraites à l'aide d'une activité de rétro-ingénierie [18].

Squidy [17] et DynaMo [2] permettent aussi l'intégration des interactions multimodales dans les applications existantes (application desktop et console de jeux). Ils utilisent des médiateurs qui convertissent les données entrées par les dispositifs d'interaction (mobile phone, manette de jeux, etc.) en des données acceptables par les dispositifs d'entrée du système en question (clavier, souris, etc.). Les médiateurs sont réalisés à l'aide des diagrammes présentant les flux de données [9][2].

Cependant, toutes ces approches sont limitées aux applications desktop ou console de jeux et ne considèrent qu'un nombre limité de modalités d'interaction.

La dernière catégorie présente un nombre réduit d'approches par rapport aux deux précédentes. Cela est dû à la difficulté d'impliquer un utilisateur final non expert dans le développement/l'évolution des applications. L'approche la plus récente dans cette catégorie est SUGILITE [21]. C'est une approche basée sur la programmation par démonstration qui permet à l'utilisateur finale d'automatiser des tâches sur une ou plusieurs applications Android. Pour automatiser une tâche, l'utilisateur lance une commande vocale et démontre la signification de cette commande à travers une manipulation directe des interfaces graphiques. Ainsi, la démonstration est enregistrée et la tâche peut être déclenchée automatiquement au lancement de la commande vocale par la suite.

À notre connaissance, il n'existe aucune approche dans la littérature qui regroupe les avantages des trois catégories présentées. C'est-à-dire, une approche qui permet à l'utilisateur final de modifier et/ou ajouter des nouvelles modalités d'interaction (ou des combinaisons de modalités) aux applications mobiles déjà installées sur son smartphone, selon ses propres préférences et besoins. Ainsi, nous proposons, dans la section suivante, une approche qui se base sur la démonstration pour l'intégration de la multimodalité sous mobile.

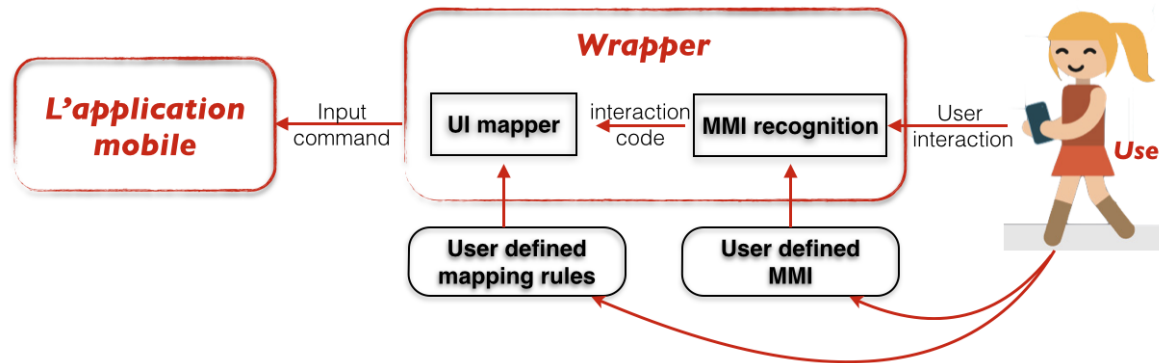
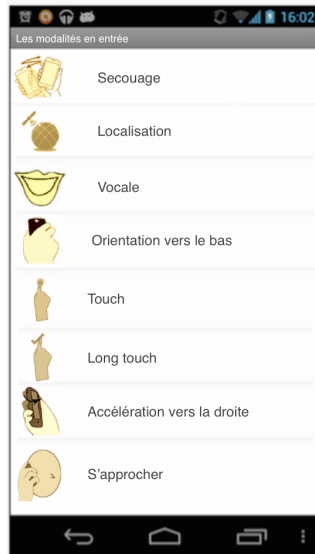


Figure 1: L'architecture de l'application

### 3 NOTRE APPROCHE

À fin de permettre aux utilisateurs finaux de changer les modalités d'interaction de leurs applications mobiles selon leurs besoins, notre approche se base sur l'utilisation d'un Wrapper. Un Wrapper est une technique de modernisation d'un système [15]. Elle permet d'éviter la modification du code source de ce dernier et de modifier indirectement ses fonctionnalités à l'aide d'un mécanisme d'encapsulation. Ce mécanisme consiste à ajouter une nouvelle couche logiciel afin de fournir une nouvelle interface et masquer la complexité de l'ancien système [15][7].

L'avantage de cette stratégie de modernisation appelée aussi « modernisation boîte noire », par rapport à la « modernisation boîte blanche » basée sur la réingénierie, est que la première implique seulement une analyse des entrées/sorties du système. Cette analyse permet de définir une interface interne et externe de Wrapper sans aucun changement de fonctionnement de l'application [15]. Par contre, l'utilisation de l'approche boîte blanche nécessite deux phases complexes [7] :



**Figure 2: Les modalités d'interaction proposées**

- L'abstraction : qui génère un modèle de l'interaction d'utilisateur à partir du code source de l'application. Cette phase se base sur une procédure de rétro-ingénierie et nécessite la présence d'un expert.
- L'ingénierie directe : qui permet d'obtenir un système opérationnel par une série de développements/modifications à partir d'un modèle abstrait (ce qui peut engendrer des risques de dysfonctionnement)

Ainsi, le Wrapper dans notre approche permet d'encapsuler les applications mobiles existantes. Il recevra les nouvelles modalités d'interaction et les simule en modalités connues par ces applications. Cela permet non seulement d'éviter le changement de code sources des applications existantes (sachant que l'utilisateur n'a pas forcément la possibilité d'accéder à leurs codes), mais aussi d'éviter le risque d'erreur de la part de l'utilisateur final qui n'est pas forcément un développeur. Notre Wrapper, comme l'illustre la Fig. 1, est composé de :

- MMI (MultiModal Interaction) recognition (la reconnaissance de l'interaction multimodale)
- UI (User Interaction) mapper (mappeur de l'interaction utilisateur)

Le premier composant «MMI recognition» fournit à l'utilisateur une interface transparente d'accès à ses applications et identifie en même temps la modalité d'interaction qu'il a utilisée. Une fois identifiée, il l'envoie vers le deuxième composant «UI mapper» qui se charge de la transformation de la modalité reconnue vers la modalité compréhensible par l'application. Ainsi, si l'application n'accepte que le tactile comme modalité d'interaction en entrée par exemple, l'utilisateur peut interagir avec une autre modalité (une interaction gestuelle par exemple) et cette dernière sera automatiquement simulée en tactile.

Cependant, le fonctionnement du Wrapper nécessite une phase de configuration qui se base, comme illustré dans la Fig. 1, sur l'interaction multimodal définie par l'utilisateur (User defined MMI) et les règles de mappage définies par l'utilisateur (User defined mapping rules). La première étape de configuration consiste à fournir à l'utilisateur la possibilité de définir les modalités d'interaction qu'il souhaite utiliser pour remplacer et/ou ajouter en concurrence des autres. Le composant «User defined MMI» propose les modalités supportées par le smartphone ainsi que les combinaisons entre ses modalités (Fig. 2).

Une fois la modalité choisie, le composant «User defined mapping rules», basé sur un système de programmation par démonstration (appelée aussi la programmation sur exemple), permet à l'utilisateur de programmer la tâche de remplacement et/ou ajout de modalité. Ce type de programmation facilite la définition de l'interface interne et externe du Wrapper pour l'utilisateur final [11][25]. Ainsi, la démonstration se réalise en manipulant directement l'interface graphique de l'application cible. Le Wrapper demandera à l'utilisateur de commencer la démonstration et déclenche une confirmation pour chaque étape. Une fois la démonstration terminée, les modifications seront enregistrées et les correspondances entre modalités (anciennes et nouvelles) seront fixées.

Supposons qu'un utilisateur souhaite remplacer la modalité tactile, pour le passage d'une page à une autre dans une application de recette de cuisine, par la modalité gestuelle «Air Browse» (Fig. 3). Il commence d'abord par lancer notre Wrapper, présentée sous forme d'une application mobile, afin de choisir la nouvelle modalité d'interaction. L'application proposera les différentes modalités disponibles (en fonction de ce qu'elle propose ainsi que des capteurs existants sur le smartphone de l'utilisateur) comme le montre la Fig. 2. L'utilisateur sélectionne la modalité «Air Browse» et valide son choix. Après validation, l'application passe à l'écran d'accueil et demande à l'utilisateur de choisir l'application à modifier (l'application des recettes) et de démontrer l'interaction à remplacer (le tactile du passage d'une page à une autre). Afin de retenir cette dernière, une confirmation est déclenchée après chaque interaction réalisée par l'utilisateur.

À la fin de la démonstration, l'application enregistre la demande et donnera, par conséquent, la possibilité d'interagir avec le «Air Browse» en équivalence avec le tactile pour passer à la page suivante des recettes.

#### 4 TRAVAUX PLANIFIÉS ET CONCLUSION

Notre approche est en cours de développement. Dans la version actuelle (sous Android), nous proposons un nombre limité de modalités d'interaction en entrée. Comme travaux futurs, nous dégagons un ensemble de perspectives répondant à cinq objectifs principaux :

- L'intégration des nouvelles modalités d'interaction : notre but est de couvrir le maximum des modalités d'interaction basées sur les capteurs en entrée. Cela augmente les choix de l'utilisateur et lui permet de bien exploiter les capteurs intégrés dans son smartphone.
- La combinaison entre les différentes modalités, selon les propriétés CARE [8] : ces propriétés permettent non seulement de rendre l'interaction plus naturelle [4], mais aussi d'augmenter la sécurité de l'interaction. Par exemple, la complémentarité entre le secouage du téléphone et le clic sur le bouton physique du volume est plus sécurisée par rapport au secouage uniquement, car ce dernier peut être involontaire.
- La gestion des conflits entre modalités : cela permet de vérifier l'ergonomie des interactions définies par l'utilisateur. On peut trouver deux types de conflits :

- (1) Conflits entre les modes d'interaction de l'utilisateur : le mode d'interaction fait référence aux cinq sens humains et aux différentes formes d'expression humaine [3]. Ainsi, les conflits peuvent être engendrés entre les modes qui utilisent les mêmes organes. Par exemple, les modes gestuel et tactile peuvent être en conflit en cas de combinaison en complémentarité ou redondance (ils utilisent presque les mêmes organes (mains, doigts, bras). Par exemple, utiliser le secouage du téléphone avec un «Long Touch» sur un bouton en complémentarité pendant une fenêtre temporelle très courte, peut être une source de conflit.

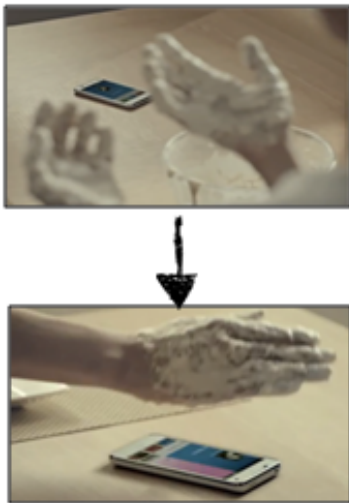


Figure 3: Un exemple de remplacement de modalité



(2) Conflits entre les modalités : les modalités qui se basent sur le même capteur physique peuvent être en conflit si elles coopèrent en concurrence.

- La gestion des erreurs : la vérification des interactions/modalités choisies par l'utilisateur est nécessaire afin d'éviter les erreurs. Par exemple, certaines interactions ne peuvent pas être remplacées par d'autres : l'écriture d'un texte avec le clavier ne peut être remplacée qu'avec la reconnaissance vocale.
- L'application des modifications/remplacements de modalité sur différentes applications. Par exemple, permettre à l'utilisateur de désactiver le vocale sur toutes les applications (mode nuit).

L'approche nécessite aussi des évaluations intensives avec les utilisateurs finaux afin de tester l'efficacité de la programmation par démonstration dans ce contexte. Nous croyons qu'une telle approche permettra une bonne exploitation à la fois des capacités humaines de communication ainsi que la puissance des smartphones. Elle ouvrira ainsi la voie vers une démocratisation de la multimodalité sous mobile.

## RÉFÉRENCES

- [1] Frédéric Aebi. 2012. Multimodal Fusion on Mobile Devices. Seminar on Multimodal Interaction on Mobile Devices. (2012). [https://diuf.unifr.ch/main/diva/sites/diuf.unifr.ch.main.diva/files/reportAebi\\_0.pdf](https://diuf.unifr.ch/main/diva/sites/diuf.unifr.ch.main.diva/files/reportAebi_0.pdf)
- [2] Pierre-Alain Avouac, Philippe Lalanda, and Laurence Nigay. 2012. Autonomic management of multimodal interaction : DynaMo in action. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, 35–44.
- [3] Yacine Bellik. 1995. *Interfaces multimodales : concepts, modèles et architectures*. Ph.D. Dissertation.
- [4] Richard A Bolt. 1980. "Put-that-there" : Voice and gesture at the graphics interface. Vol. 14. ACM.
- [5] Jullien Bouchet, Laurence Nigay, and Thierry Ganille. 2004. ICARE software components for rapidly developing multimodal interfaces. In *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 251–258.
- [6] David F Carr. 1998. Web-enabling legacy data when resources are tight. *Internet World (1998)*.
- [7] Santiago Comella-Dorda, Kurt C Wallnau, Robert C Seacord, and John E Robert. 2000. A Survey of Black-Box Modernization Approaches for Information Systems.. In *icsm*. 173–183.
- [8] Joëlle Coutaz and Laurence Nigay. 1994. Les propriétés CARE dans les interfaces multimodales. In *IHM*, Vol. 94. 7–14.
- [9] Fredy Cuenca, Jan Van den Bergh, Kris Luyten, and Karin Coninx. 2016. Hasselt : Rapid Prototyping of Multimodal Interactions with Composite Event-Driven Programming. *International Journal of People-Oriented Programming (IJPOP)* 5, 1 (2016), 19–38.
- [10] Rachel Demumieux and Patrick Losquin. 2005. Gather customer's real usage on mobile phones. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*. ACM, 267–270.
- [11] Fabrice Depaulis, Laurent Guittet, and Christophe Martin. 2003. Apprends ce que je fais. In *15ème Conférence Francophone sur l'Interaction Homme-Machine*. 236–239.
- [12] Nadia Elouali. 2014. *Approche à base de modèles pour la construction d'applications mobiles multimodales*. Ph.D. Dissertation. Lille 1.
- [13] Nadia Elouali, Xavier Le Pallec, José Rouillard, and Jean-Claude Tarby. 2014. MIMIC : leveraging sensor-based interactions in multimodal mobile applications. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2323–2328.

- [14] Nadia Elouali, Xavier Le Pallec, José Rouillard, and Jean-Claude Tarby. 2014. A Model-based Approach for Engineering Multimodal Mobile Interactions. In *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*. ACM, 52–61.
- [15] A Sivagnana Ganesan and T Chithralekha. 2016. A Survey on Survey of Migration of Legacy Systems. In *Proceedings of the International Conference on Informatics and Analytics*. ACM, 72.
- [16] Michael Johnston. 2009. Building multimodal applications with EMMA. In *Proceedings of the 2009 international conference on Multimodal interfaces*. ACM, 47–54.
- [17] Werner A König, Roman Rädle, and Harald Reiterer. 2010. Interactive design of multimodal user interfaces. *Journal on Multimodal User Interfaces* 3, 3 (2010), 197–213.
- [18] Fabrizio Lamberti and Andrea Sanna. 2008. Extensible GUIs for remote application control on mobile devices. *IEEE computer graphics and applications* 28, 4 (2008).
- [19] Fabrizio Lamberti, Andrea Sanna, Gilles Carlevaris, and Claudio Demartini. 2015. Adding pluggable and personalized natural control capabilities to existing applications. *Sensors* 15, 2 (2015), 2832–2859.
- [20] Fabrizio Lamberti, Andrea Sanna, Gianluca Paravati, and Claudio Demartini. 2013. Endowing existing desktop applications with customizable body gesture-based interfaces. In *Consumer Electronics (ICCE), 2013 IEEE International Conference on*. IEEE, 558–559.
- [21] Toby Jia-Jun Li, Amos Azaria, and Brad A Myers. 2017. SUGILITE : creating multimodal smartphone automation by demonstration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 6038–6049.
- [22] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, Murielle Florins, and Daniela Trevisan. 2004. Usixml : A user interface description language for context-sensitive user interfaces. In *Proceedings of the ACM AVI'2004 Workshop "Developing User Interfaces with XML : Advances on User Interface Description Languages"(Gallipoli)*. 55–62.
- [23] Andreas Möller. 2015. *Leveraging Mobile Interaction with Multimodal and Sensor-Driven User Interfaces*. Ph.D. Dissertation. Technical University Munich.
- [24] Andreas Möller, Stefan Diewald, Luis Roalter, and Matthias Kranz. 2014. Supporting Mobile Multimodal Interaction with a Rule-Based Framework. *arXiv preprint arXiv :1406.3225* (2014).
- [25] II MOTIVATIONS. 2006. An end-user programming paradigm for pervasive computing applications. (2006).
- [26] Tiago Alexandre Custódio Reis. 2008. *Universally accessible mobile multimodal artefacts*. Ph.D. Dissertation.
- [27] Marcos Serrano, Laurence Nigay, Rachel Demumieux, Jérôme Descos, and Patrick Losquin. 2006. Multimodal interaction on mobile phones : development and evaluation using ACICARE. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, 129–136.

